TESIS - KS142501

PENGAMBILAN KONTEN UTAMA PADA WEBSITE PEMERINTAH DAERAH MENGGUNAKAN PENDEKATAN TEMPLATE-BASED DAN KLASIFIKASI NAIVE-BAYES

FAJARA KURNIAWAN N.H.

NRP. 05211550012003

DOSEN PEMBIMBING:

NUR AINI RAKHMAWATI, MSc. Eng, Ph. D

NIP. 198201202005012001

PROGRAM MAGISTER

DEPARTEMEN SISTEM INFORMASI

FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom)
di
Institut Teknologi Sepuluh Nopember

Oleh :
Fajara Kurniawan Nasrullah Hariyadi
NRP. 05211550012003

Tanggal Ujian : 10 Juli 2018
Periode Wisuda : September 2018

Disetujui Oleh :

1. Nur Aini Rakhmawati, M.Sc.Eng, Ph.D
   NIP. 19820120 2005012001 (Pembimbing 1)

2. Dr. Ir. Aris Tjahyanto, M.Kom
   NIP. 19650310 1991021001 (Penguji 1)

3. Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom
   NIP. 19730219 1998021001 (Penguji 2)

Dekan
Fakultas Teknologi Informasi dan Komunikasi

Dr. Agus Zainal Arifin, S.Kom., M.Kom
NIP. 19720809 199512 1 001

halaman ini sengaja dikosongkan

Pengambilan Konten Utama pada Website Pemerintah Daerah Menggunakan Pendekatan Template-based dan Klasifikasi Naïve-Bayes

Nama Mahasiswa	:	Fajara Kurniawan N.H.
NRP	:	05211550012003
Dosen Pembimbing	:	Nur Aini Rakhmawati, MSc. Eng, Ph.D

ABSTRAK

Internet dan *World Wide Web* dapat memberikan kemampuan penting yang dapat mendorong kemampuan yang dimiliki oleh pemerintahan. Hal tersebut adalah kemampuan yang dapat membuat pemerintah lokal dapat mendistribusikan informasi serta warga negara dapat menerima informasi terkini mengenai urusan pemerintah lokal dengan biaya yang murah dan mudah. Hal ini lebih sering disebut dengan E-Government. Penggunaan E-Government di Indonesia sendiri sudah didukung penuh oleh instruksi Presiden Republik Indonesia.

Egovbench adalah aplikasi *monitoring* dan pengukuran performa dari website dan media sosial resmi dari pemerintah daerah di Indonesia. Untuk melakukan tugasnya tersebut egovbench harus melakukan proses pengambilan informasi ke setiap halaman web pada setiap situs web resmi yang dimiliki oleh pemerintah daerah.

Setiap halaman web yang ada akan memiliki sebuah *main content*. *main content* adalah sebuah bagian, segmen atau blok yang berisi konten yang berupa teks atau dalam bentuk multimedia yang berada pada sebuah halaman web yang bukan merupakan halaman web landing atau beranda dan bersifat unik pada satu halaman web. Informasi-informasi penting mengenai pemerintahan daerah umumnya berada di dalam textitmain content sehingga diperlukan *web content extractor* untuk mengambil informasi-infomrasi tersebut.

Pada penelitian ini, untuk mengatasi permasalahan mengenai pengambilan *main content* tersebut, dilakukan penggabungan antara dua pendekatan yang telah ada yaitu pendekatan *template-based* dan pendekatan machine learning dengan menggunakan Naïve-Bayes *Classifier*. Umumnya penelitian terdahulu yang dilakukan masih menggunakan satu tipe pendekatan yaitu antara menggunakan pendekat-

an *template-based* atau menggunakan *machine learning*. Kontribusi dari penelitian ini adalah mengenai bagaimana hasil dari pengambilan textitmain content dengan menggunakan gabungan dua pendekatan antara pendekatan *template-based* dan pendekatan Klasifikasi Naïve-Bayes.

Tantangan yang dihadapi pada penelitian ini adalah bagaimana struktur halaman web yang dimiliki oleh pemerintah daerah dapat menyulitkan pada tahap pengambilan *main content* dengan pendekatan *template-based* terutama efek dari *Content Management System* (CMS) pada struktur halaman web. Hasil yang didapatkan memperlihatkan bahwa dengan menggunakan bahwa dengan menggunakan gabungan dua tipe pendekatan dapat memberikan hasil yang lebih akurat dalam memprediksi kategori halaman web dengan akurasi yang dicapai sebesar 68% dibandingkan pendekatan yang digunakan saat ini pada egovbench dengan akurasi sebesar 59%.

Kata kunci: *web content extractor, template-based, machine learning*.

Extracting Main Content on Local Government Website Using Template-Based Approach and Naïve-Bayes Classification

Name	:	Fajara Kurniawan N.H.
NRP	:	05211550012003
Supervisor	:	Nur Aini Rakhmawati, MSc. Eng, Ph.D

ABSTRACT

The Internet and the World Wide Web offer capabilities that can help increase government capabilities further. It is a capability that enables local governments to distribute information and citizens can receive up-to-date information about local government affairs at low cost and convenience. These activities is commonly referred as E-Government. The use of E-Government in Indonesia itself is fully supported by the instruction of the President of the Republic of Indonesia.

Egovbench Is a monitoring and performance measurement application of official website and social media from local government in Indonesia. For egovbench do this task, egovbench need to take information on every web page on every official website of local government.

Every web page will have a main content. Main content is a section, segment or block that contains text or multimedia on single web page that is not a landing page of web site or homepage and is unique to single web page. Important information about local governance generally lies within main content thus the need of web content extractor to extract that information.

In this research, we combine the two approaches that already existed, template-based approach and machine learning approach using Naïve-Bayes Classifier, to solve the problem of extracting main content from the webpage. Generally, previous research that has been conducted is using one type of approach, it is either using a template-based approach or using machine learning approach. The contribution of this research is on how the results of the main content extraction using a combination of two approaches between the template-based approach and the Naïve-Bayes Classification approach.

The challenge that this research faced is mainly come from the structure of web

pages in official website owned by local government which hamper the template-based approach especially the effect of Content Management System on web page structures. The results show that using a combination of two types of approaches can yield more accurate results in predicting web page categories with an accuracy of 68% compared to current approach in egovbench with an accuracy of 59%.

Keywords: *web content extractor, template-based, machine learning*.

KATA PENGANTAR

Syukur Alhamdulillah dipanjatkan oleh peneliti atas segala petunjuk, pertolongan, kasih saying, dan kekuatan Allah SWT berikan. Hanya karena ridho-Nya, peneliti dapat menyelesaikan laporan penelitian tesis yang berjudul **Pengambilan Konten Utama pada Website Pemerintah Daerah Menggunakan Pendekatan Template-based dan Klasifikasi Naïve-Bayes**.

Terima kasih terucap untuk seluruh pihak yang sangat luar biasa dalam memantu penelitian ini, yaitu:

1. Keluarga penulis yang senantiasa mendoakan, mendukung, dan mendorong penulis untuk menyelesaikan penelitian ini.
2. Ibu Nur Aini Rakhmawati, MSc. Eng, Ph.D, selaku dosen pembimbing sekaligus dosen wali yang telah meluangkan waktu, tenaga, dan pikiran untuk membimbing, mendukung dan mendoakan dalam penyelesaian penelitian ini
3. Seluruh bapak ibu dosen dan karyawan di Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember.
4. Teman-teman S2 Sistem Informasi yang telah menemani suka dan duka penulis dalam menyelesaikan masa perkuliahan dan penelitian ini.
5. Teman-teman Laboratorium Akuisisi Data dan Disseminasi Informasi yang telah juga memberikan semangat dan membantu dalam penyelesaian penelitian ini.

Penyusunan laporan ini masih jauh dari sempurna, untuk itu peneliti menerima kritikan dan saran yang membangun untuk perbaikan di masa mendatang. Penelitian ini diharapkan dapat menjadi salah satu acuan bagi penelitian yang serupa dan bermanfaat bagi pembaca.

Surabaya, 27 Juli 2018

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

Halaman ini sengaja dikosongkan

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR KODE

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Bab ini terdiri menjelaskan mengenai latar belakang dilakukannya penelitian, perumusan masalah, tujuan dan kontribusi penelitian, batasan penelitian, dan sistematika penulisan.

1.1 Latar Belakang

Internet dan *World Wide Web* memberikan dua hal penting yang dapat mendorong kemampuan yang dimiliki oleh pemerintahan lokal [Musso et al., 2000] . Pertama, pemerintah lokal dapat mendistribusikan informasi serta warga negara dapat menerima informasi terkini mengenai urusan pemerintah lokal dengan biaya yang murah dan mudah. Distribusi informasi berbasis internet dapat memperbaiki pengetahuan warga negara secara signifikan dikarenakan adanya kemudahan akses, ketersediaan informasi yang konstan, dan kemampuan untuk mempresentasikannya dalam format visual yang menyenangkan dan mudah dipahami. Kedua, melalui *e-mail* dan *chat room*, Internet memfasilitasi komunikasi jarak jauh, tanpa halangan waktu, dan tanpa melihat kelompok dan institusi sosial yang berbeda.

Secara umum penerapan Teknologi informasi dan komunikasi (TIK) di pemerintahan lebih sering disebut dengan layanan *E-Government*. Friedman [Friedman and Bryen, 2007] mengatakan bahwa masing-masing negara dalam mengembangkan situs web *e-government* tidak harus bergantung pada pedoman dan standar industri, namun harus menetapkan standar atau undang-undang mereka sendiri. Penggunaan *E-Government* di Indonesia sendiri sudah didukung penuh oleh Instruksi Presiden Republik Indonesia no.3 Tahun 2003 tentang kebijakan dan strategi Nasional Pengembangan E-Government [dan Informatika, 2003]. Penggunaan *E-Government* merupakan upaya untuk mengembangkan penyelenggaraan pemerintahan yang berbasis elektronik dalam rangka meningkatkan kualitas layanan publik secara efektif dan efisien [Dewi and Mudjahidin, 2014]

Situs web pemerintah menjadi hal yang penting dalam menarik warga negara untuk menggunakan *e-government* dan memperbaiki persepsi stakeholder eks-

ternal terhadap pemerintah. Situs web yang buruk dapat membatasi aksesibilitas dan kegunaan dimana pada akhirnya mengikis kredibilitas [Huang and Benyoucef, 2014, Youngblood and Mackiewicz, 2012]. Selain itu, hal-hal seperti desain, fungsionalitas, dan konten situs web dapat mempengaruhi daya tarik dan pengalaman pengguna secara online terhadap pemerintah [Feeney and Brown, 2017].

Berdasarkan data pada situs web Kementerian Komunikasi dan Informatika (www.kominfo.go.id), wilayah Indonesia mempunyai jumlah pemerintahan 548 dengan rincian 34 provinsi, 416 kabupaten, 98 kota. Pemerintah daerah yang memiliki situs web resmi sebanyak 485, dimana terjadi peningkatan dari data sebelumnya pada penggunaan situs web resmi pemerintahan di Indonesia[Hermawan, 2015]. Penelitian sebelumnya yang dilakukan oleh Dana Sulistyo K menemukan bahwa setelah melalui beberapa rangkaian penghitungan situs web resmi pemerintahan tersebut masih belum memenuhi kriteria yang tepat [Sulistyo et al., 2008]. Dari Hanif Hoesin didapatkan bahwa masih banyak situs web resmi pemerintahan yang bahkan dinilai tidak aktif [Hoesin et al., 2008]. Oleh karena itu dibutuhkan aplikasi yang dapat melakukan penilaian secara *real time* dengan cara perankingan sehingga setiap daerah akan berusaha untuk memberikan yang terbaik.

Berangkat dari permasalahan tersebut maka dikembangkanlah perangkat lunak untuk *monitoring* dan pengukuran performa dari situs web resmi dan media sosial resmi dari pemerintah dengan nama egovbench. Dengan egovbench performa dan rangking tiap situs web resmi dan media sosial resmi pemerintah akan terlihat. Dalam pelaksanaannya egovbench terdiri atas berberapa tahap yang termasuk dalam roadmap pengembangan egovbench. Tahapan dari roadmap egovbench sendiri meliputi (1) *crawling data* dan ranking. (2) *content detection* dan *storage*. (3) *data integration* dan *text summarization*. (4) *responsive of social media*. Untuk saat ini egovbench sendiri berada pada tahap satu yang mana selanjutnya akan dikembangkan menuju tahap kedua. Untuk melakukan tugasnya tersebut, Egovbench perlu untuk melakukan pengambilan informasi dari situs web resmi yang dimiliki oleh pemerintah daerah di Indonesia.

Pada setiap halaman web akan terdiri atas *main content* dan *noisy content*. Se-

cara harfiah *main content* didefinisikan sebagai informasi utama yang ada pada sebuah halaman web, dimana *noisy content* didefinisikan sebagai informasi-informasi yang tidak berkaitan dengan informasi utama seperti komentar, menu navigasi, iklan dan konten-konten lain. Untuk mendapatkan informasi yang diinginkan maka egovbench perlu untuk mengambil *main content* dari halaman web yang ada pada situs web resmi pemerintah daerah di Indonesia.

Untuk saat ini pada egovbench dalam melakukan pengambilan informasi pada proses *web crawling* pada halaman web, egovbench menggunakan pendekatan kesamaan baris pada setiap halaman untuk *keyword* yang dicari. Pada pendekatan tersebut, egovbench menghilangkan semua *tag* HTML ketika pengambilan halaman web dilakukan, setelah semua *tag* dihilangkan maka yang tersisa kemudian hanyalah konten dari setiap tag, setelah itu dilakukan komparasi untuk setiap baris yang ada. Jika pada suatu baris memiliki konten yang sama pada berberapa halaman web maka kemungkinan besar konten tersebut bukanlah *main content* yang kemudian baris tersebut akan dihilangkan. Kelemahan pendekatan ini adalah karena hanya membandingkan isi konten pada setiap baris pada setiap halaman tanpa memperhatikan *tag* maka dapat memberikan kesalahan pengambilan informasi dikarenakan umumnya *main content* akan berada pada *tag-tag* spesifik yang menjadi tempat *main content* berada sehingga dengan menghapus *tag* maka bisa menimbulkan kesalahan pengambilan informasi karena seluruh konten memiliki bobot yang sama tanpa menghiraukan tag dimana konten tersebut berada.

Secara umum manusia dapat mengenali dan membedakan mengenai *main content* dari sebuah halaman web berdasarkan pengetahuan, pengalaman dan intuisi mereka, akan tetapi untuk *automated information extractors*, hal ini menjadi sebuah tantangan yang besar. Hal ini dikarenakan banyak halaman web yang memiliki format yang berbeda untuk setiap situs web [Louvan, 2009, Yunis, 2016]. Untuk mengatasi permasalahan tersebut, berbagai penelitian yang berfokus untuk membangun *web content extractor*. Berbagai pendekatan telah banyak dilakukan diantaranya menggunakan pendekatan visual dengan melihat letak *main content* dengan petunjuk visual, menggunakan pendekatan *machine learning* pada konten yang ada

pada halaman web terutama konten yang bersifat teks, atau menggunakan pendekatan *template-based* yang melihat kesamaan konten pada halaman web yang mirip.

Berberapa penelitian yang menggunakan pendekatan *machine learning* untuk melakukan pencarian *main content* umumnya membagi halaman web menjadi sebuah blok atau bagian yang kemudian pada setiap blok atau bagian tersebut dilakukan pendekatan *machine learning* untuk menentukan apakah blok tersebut merupakan *main content* atau tidak. Salah satu penelitian yang menggunakan pendekatan ini adalah penelitian yang dilakukan oleh Kohlschütter [Kohlschütter et al., 2010] dan Yao et al. [Yao and Zuo, 2013].

Kohlschütter menggunakan *wrapper* untuk membentuk blok-blok dimana blok-blok tersebut akan dilakukan pendekatan *machine learning* untuk menentukan *main content*. Yao juga menggunakan definisi dan teknik yang digunakan oleh Kohlschütter untuk membagi halaman web menjadi blok dan menambahkan berberapa fitur-fitur lain untuk menentukan *main content* dengan menggunakan *machine learning*. Dengan melihat penelitian-penelitian yang telah dilakukan[Kohlschütter et al., 2010, Lundgren et al., 2015, Weninger et al., 2010, Yao and Zuo, 2013], pendekatan *machine learning* selain bergantung dengan proses *machine learning* sendiri juga sangat bergantung dengan proses *wrapper* untuk mengolah halaman web untuk dilakukan proses *machine learning*.

Pendekatan berbasis visual berfokus kepada analisis fitur visual dari isi dokumen seperti yang dirasakan oleh pembaca manusia [Zeleny et al., 2017]. Pendekatan Visual ini pertama kali diperkenalkan oleh Cai [Cai et al., 2003] dengan memperkenalkan teknik *Vision-Based Page Segmentation Algorithm* (VIPS) untuk mendeteksi konten yang ada pada halaman web. VIPS mencoba untuk mensimulasikan pendekatan berdasarkan pengguna manusia untuk memahami struktur konten dari sebuah halaman web.

Chen [Chen et al., 2003] membagi halaman web menjadi bagian-bagian berdasarkan visual yang dinamakan *Explicit Separator Detection*. *Explicit Separator Detection* membagi halaman web menjadi berberapa partisi atau bagian. Contohnya Chen mendefinisikan bahwa *left side bar* adalah satu perempat bagian dari halaman

web pada bagian kiri dan *right side bar* adalah satu perempat bagian dari halaman web pada bagian kanan.

Pendekatan *template-based* bertujuan untuk mengidentifikasi dan menyaring bagian-bagian dari sebuah halaman web yang berulang kali muncul pada halaman yang mirip [Zeleny et al., 2017]. *Template* dapat didefinisikan sebagai tata letak halaman web dengan tempat atau slot dimana isi variabel dapat dimasukkan [Yunis, 2016].

Penelitian yang menggunakan pendekatan template-based diantaranya adalah penelitian yang dilakukan oleh Yossef [Bar-Yossef and Rajagopalan, 2002]. Yossef mengajukan teknik *Largest Pagelet* dalam pendeteksian *template*. Teknik *Largest Pagelet* sendiri didasarkan pada teknik teknik *shingle/shingling* dimana *pagelet* yang memiliki nilai lebih dari batas yang ditentukan dianggap sebagai sebuah template. Gibson [Gibson et al., 2007] mengajukan teknik *template detection* dengan melakukan *hashing* pada tiap tag yang ada pada halaman web. Dengan melihat frekuensi kemunculan hash maka template dapat dibentuk. Alarte [Alarte et al., 2015] mengembangkan sebuah *tool* untuk melakukan ekstraksi *template* yang dinamakan *Template Extractor*. Untuk melakukan pencarian *template* dari sebuah halaman web, TeMex mencari terlebih dahulu kandidat halaman web yang memiliki kesamaan template dengan halaman web yang ingin dicari templatenya yang kemudian dibuat sebuah *key page*. *Key page* dibuat dengan melakukan komparasi struktur DOM dengan pendekatan *Equal Top-Down Mapping*. Dari *Key Page* yang telah dilakukan komparasi tersebut dibentuk *template* dari halaman web. Dengan melihat penelitian-penelitian yang telah dilakukan, pendekatan menggunakan *template-based* sendiri membutuhkan sekumpulan dari halaman web yang memiliki kemiripan dan tidak dapat dilakukan pada satu halaman web saja.

Penelitian-penelitian yang telah dilakukan sebelumnya tersebut umumnya berfokus hanya kepada satu pendekatan dan sejauh ini masih belum ada penelitian yang mencoba menggabungkan berberapa pendekatan sekaligus. Pada penelitian ini akan diajukan sebuah teknik *web content extractor* dengan menggunakan pendekatan gabungan yang menggunakan penggabungan antara pendekatan *template-based* dan

machine learning.

Pemilihan penggunaan metode gabungan ini didasarkan bahwa secara umum pada setiap halaman web pada situs web resmi pemerintah daerah di Indonesia akan memiliki kesamaan pola dimana pola-pola tersebut umumnya merupakan sebuah *noisy content* seperti menu navigasi atau *footer*. Berdasarkan kesamaan pola pada setiap halaman web pada sebuah situs web resmi pemerintah daerah maka dapat dilakukan pendekatan *template-based* dalam melakukan pencarian *main content*. Pada pendekatan *template-based* sendiri didasarkan pada prinsip setiap halaman web pada sebuah situs web akan memiliki kesamaan fitur,pola atau ciri khas yang dinamakan *template* dimana umumnya *template* sendiri berisi hal-hal seperti *noisy content* seperti contohnya menu navigasi atau *footer*.

Dengan memanfaatkan pendekatan *template-based* untuk menemukan pola-pola yang merupakan *noisy content*, maka bagian yang telah tersaring adalah kandidat sebagai *main content*. Selanjutnya untuk menentukan apakah bagian tersebut merupakan *main content* atau bukan *main content*, digunakan pendekatan *machine learning*.

Penggunaan pendekatan *machine learning* untuk menentukan sebuah kandidat *main content* sebagai *main content* atau tidak, didasarkan pada regulasi yang diterbitkan oleh Kementrian Komunikasi dan Informasi Republik Indonesia dalam Panduan Penyelenggaraan Situs Pemerintah Daerah yang berisi mengenai informasi yang harus ada pada situs web pemerintah daerah di Indonesia. Dengan mengacu pada hal tersebut maka dapat dikembangkan sebuah pendekatan *machine learning* untuk menentukan *main content* dari kandidat *main content*. Dengan penggunaan pendekatan gabungan ini maka diharapkan dapat meningkatkan akurasi dalam penentuan *main content* pada halaman web pada situs web pemerintahan daerah.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, berberapa poin yang perlu digaris bawahi adalah (1) Perlu adanya *web content extractor* yang dapat meningkatkan proses akurasi pengambilan *main content* pada egovbench (2) Pengambilan *main*

content sekarang menggunakan *resource* yang besar dan waktu yang cukup lama dalam melakukan tugasnya. Maka, rumusan masalah yang ingin dijawab melalui penelitian ini diuraikan sebagai berikut:

1. Bagaimana formulasi pendekatan gabungan dalam pengambilan *main content* yang diajukan dalam aplikasi egovbench untuk mengambil *main content* pada situs web resmi pemerintah daerah di Indonesia ?
2. Bagaimana kinerja pendekatan gabungan yang diajukan dalam pengambilan *main content* pada situs web resmi pemerintah daerah di Indonesia ?

1.3 Tujuan dan Manfaat Penelitian

Sesuai dengan perumusan masalah yang ada, maka tujuan penelitian ini adalah menghasilkan sebuah usulan pendekatan mengenai pengambilan *main content* yang spesifik ditujukan untuk mengambil *main content* yang ada pada situs web resmi pemerintah daerah di Indonesia.

Manfaat dari penelitian ini adalah memberikan sebuah usulan pengambilan *main content* pada khusus untuk situs web resmi pemerintah daerah di Indonesia.

1.4 Kontribusi Penelitian

Penelitian ini dapat memberikan kontribusi secara teoritis maupun secara praktis

1.4.1 Kontribusi Teoritis

Kontribusi secara teori diperoleh dari (1) usulan pengambilan *main content* yang spesifik ditujukan untuk mengambil *main content* yang ada pada situs web resmi pemerintah daerah di Indonesia. (2) Memberikan gambaran mengenai penggunaan metode gabungan pendekatan *template-based* dan pendekatan Klasifikasi Naïve-Bayes dalam proses pengambilan *main content* (3) Memberikan gambaran mengenai pengaruh *Content Management System* dalam menyimpan *main content* dan menyajikan *main content* pada sebuah halaman web dan pengaruhnya terhadap hasil yang dicapai oleh pendekatan *template-based* dalam pengambilan *main content*.

1.4.2 Kontribusi Praktis

Kontribusi secara praktis pada penelitian ini adalah (1) Meningkatkan proses *web crawling* pada egovbench terutama dalam akurasi pengambilan *main content*. (2) Memberikan gambaran mengenai bagaimana perkembangan situs web resmi pemerintahan daerah di Indonesia (3) Memberikan gambaran mengenai karakteristik-karakteristik yang ada pada situs web resmi pemerintahan daerah di Indonesia terutama mengenai struktur halaman web yang dimiliki oleh situs web resmi pemerintah daerah dan bagaimana *main content* ditampilkan pada halaman web di situs web resmi pemerintah daerah. (4) Memberikan gambaran mengenai bagaimana pengambilan *main content* dapat meningkatkan akurasi dalam memprediksi kategori halaman web sesuai dengan Panduan Penyelenggaraan Situs Pemerintah Daerah.

1.5 Keterbaruan(*Novelty*)

Berdasarkan penyusunan penelitian dari pendahuluan, perumusan masalah, tujuan penelitian dan manfaat penelitian dapat ditentukan keterbaruan (novelty) penelitian ini. Pada Penelitian sebelumnya mengenai *web content extractor* umumnya adalah menggunakan satu macam pendekatan dalam menentukan *main content* dimana setiap masing-masing pendekatan mempunyai keunggulan dan kelemahannya masing-masing. Pada penelitian ini akan diusulkan menggunakan pendekatan gabungan dimana pendekatan *template-based* akan digabungkan dengan pendekatan *machine learning*. Pada penelitian ini, pendekatan *template-based* digunakan untuk menghilangkan segmen atau blok yang sering muncul (*noisy content*) pada halaman web yang ada pada sebuah situs web. Untuk melakukan hal tersebut diajukan *Multiple Restricted Top-Down Mapping*, dimana algoritma ini akan memulai perbandingan dengan memulai dari *tag* paling atas terlebih dahulu kemudian berfokus pada *tag* dibawahnya yang dianggap bukan kandidat *template*. Dengan hanya melakukan perbandingan pada *node* yang secara posisi berada di bagian atas maka tidak perlu dilakukan perbandingan untuk *node* yang berada dibawah node yang dianggap sebagai kandidat *template*. Kemudian dengan menghilangkan segmen atau blok yang bukan merupakan *template*, segmen atau blok yang tersisa akan diang-

gap sebagai kandidat *main content* akan diproses dengan menggunakan pendekatan *machine learning* untuk menentukan apakah blok atau segmen tersebut merupakan *main content* atau bukan *main content*. Dengan menggunakan pendekatan *machine learning* yang disesuaikan dengan pendefinisian *main content* dan *noisy content* yang spesifik mengenai situs web resmi pemerintah daerah maka diharapkan dapat meningkatkan akurasi dalam pengambilan *main content*.

1.6 Batasan Penelitian

Penelitian ini memiliki ruang lingkup yang akan menjadi batasan dalam penelitian ini. Batasan penelitian ini antara lain:

1. Data situs web yang digunakan adalah data situs web resmi pemerintahan daerah di seluruh Indonesia dan hanya situs web yang memiliki domain go.id.
2. Resource yang digunakan adalah *resource* yang digunakan pada egovbench saat ini.
3. Halaman beranda yang dimaksud pada penelitian ini adalah halaman *landing* dari *link url* resmi pemerintah daerah dan memiliki setidaknya tujuh buah *link* yang mengarah pada *domain* yang sama dengan *url* resmi untuk setiap situs web resmi pemerintah daerah (pengecualian untuk www dimana www.surabaya.go.id dianggap sama dengan surabaya.go.id) dengan menghiraukan *link* yang mengarah ke *domain* lain termasuk link yang mengarah ke *subdomain* yang berada dibawah *url* resmi pemerintah daerah tersebut.
4. *crawling link* pada situs web resmi pemerintah daerah hanya dilakukan pada *domain* utama sesuai *link url* resmi pemerintah daerah
5. Pada penelitian ini *main content* adalah konten yang berada di dalam tag <body> dan tag <body> setidaknya memiliki minimum satu buah *child node* atau tag html yang berada tepat satu level di bawah tag <body>

1.7 Sistematika Penulisan

Sistematika penulisan laporan proposal penelitian ini adalah sebagai berikut :

1. **Bab 1 Pendahuluan**

Bab ini berisi pendahuluan yang menjelaskan latar belakang permasalahan, perumusan masalah, tujuan penelitian, manfaat penelitian, kontribusi penelitian, batasan penelitian serta sistematika penulisan.

2. **Bab 2 Kajian Pustaka**

   Bab ini berisi kajian terhadap teori dan penelitian-penelitian yang sudah ada sebelumnya. Kajian pustaka ini bertujuan untuk memperkuat dasar dan alasan dilakukan penelitian.

3. **Bab 3 Metodologi Penelitian**

   Bab ini berisi mengenai rancangan penelitian, lokasi dan tempat penelitian, serta tahapan-tahapan sistematis yang digunakan selama melakukan penelitian.

4. **Hasil dan Pembahasan**

   Bab ini berisi mengenai penjelasan mengenai temuan, hasil dan pembahasan dari hasil penelitian yang dilakukan pada penelitian ini.

5. **Kesimpulan dan Saran**

   Bab ini berisi mengenai kesimpulan dan saran yang didapatkan dari penelitian yang dilakukan beserta dengan referensi mengenai penelitian kedepannya

6. **Daftar Pustaka**

   Berisi daftar referensi yang digunakan dalam penelitian ini, baik jurnal, buku, maupun artikel.

BAB 2

KAJIAN PUSTAKA

Bab ini menjelaskan mengenai teori-teori yang digunakan dalam penyusunan thesis serta kajian pustaka yang diambil dari penelitian-penelitian sebelumnya yang relevan. Kajian pustaka ini selanjutnya akan dibangun sebagai landasan dalam melakukan penelitian ini.

2.1 Kajian Teori

Pada bagian ini akan dijelaskan mengenai teori-teori yang terkait dengan penelitian yang akan dilakukan.

2.1.1 *E-Government*

Electronic Government adalah aplikasi teknologi informasi yang berbasis internet dan perangkat lainnya yang dikelola oleh pemerintah untuk keperluan penyampaian informasi dari pemerintah kepada masyarakat, mitra bisnisnya, dan lembaga-lembaga lain secara online [Sosiawan and Arief., 2008] . Mulus [Mulus, 2009] menyatakan bahwa teknologi informasi sangat berhubungan dengan kondisi internal yang baik akan dipakai oleh sistem pemerintahan yang bermanfaat untuk menyampaikan informasi dan pelayanan yang diperuntukkan bagi masyarakat yang akan mengurus kepentingan bisnis atau yang lainnya . Jadi dapat disimpulakan bahwa E-Government adalah sebuah teknologi informasi yang digunakan untuk membantu sistem pemerintahan dalam melayani masyarakat ataupun bisnisnya agar lebih baik.

Kementrian Komunikasi dan Informasi Republik Indonesia mendefinisikan *electronic government* sebagai aplikasi teknologi informasi yang berbasis internet dan perangkat lainnya yang dikelola oleh pemerintah untuk keperluan penyampaian informasi dari pemerintah kepada masyarakat, mitra bisnisnya, dan lembaga-lembaga lain secara online [Sosiawan and Arief., 2008].

Dari pengertian diatas dapat disimpulkan bahwa *E-government* merupakan proses pemanfaatan teknologi informasi di pemerintahan dengan tujuan sebagai alat

Gambar 2.1: Tampilan Egovbench

untuk membantu menjalankan sistem pemerintahan agar lebih efisien, efektif, dan produktif. Salah satu bagian dari *E-government* adalah memiliki situs web dan media sosial resmi yang berkualitas dan bermutu agar masyarakat dapat mengetahui mengenai pemerintahan tersebut.

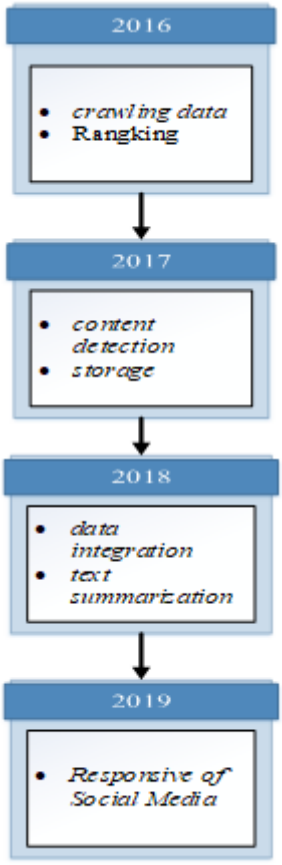
2.1.2 E-Govbench

Egovbench adalah sebuah aplikasi berbasis web yang melakukan perangkingan terhadap situs web dan sosial media yang dimiliki oleh pemerintah daerah. Egovbench menilai apakah situs web dan sosial media tersebut benar-benar digunakan sebagai media E-Government yang melayani masyarakat atau tidak. Egovbench digunakan untuk mengetahui kualitas, performa dan mutu dari situs web dan media sosial resmi pemerintah seperti yang terlihat pada gambar 2.1

Untuk menilai kualitas dan performa dari situs web dan media sosial sendiri, Egovbench memiliki berberapa kriteria yang didapatkan berdasarkan instruksi presiden untuk penerapan situs E-Government sendiri [dan Informatika, 2003]. Berikut adalah penjelasan mengenai faktor yang menjadi kriteria penilaian pada Egovbench

[dan Informasi, 2009]:

1. Kelengkapan situs web
    - Selayang Pandang adalah menjelaskan secara singkat keberadaan dan informasi Pemerintahan seperti sejarah, motto daerah, lambang dan arti lambang, lokasi dalam bentuk peta, visi dan misi
    - Pemerintahan Daerah menjelaskan struktur organisasi yang ada dipemerintahan daerah bersangkutan (eksekutif, legislatif) beserta nama, alamat, telepon, email dari pejabat daerah. Sehingga akan dinilai dari profil pemerintahan daerah mereka, Profil Pemimpin, Struktur Organisasi.
    - Geografi menjelaskan keterangan keadaan pada lokasi daerah yaitu meliputi topografi, demografi, cuaca dan iklim, sosial dan ekonomi, budaya.
    - Peta Wilayah dan Sumberdaya menyajikan batas administrasi wilayah dalam bentuk peta wilayah dan juga sumberdaya dalam bentuk peta sumberdaya
    - Peraturan/Kebijakan Wilayah menjelaskan Peraturan Daerah (Perda) yang dikeluarkan oleh daerah bersangkutan.
    - Berita dari lingkungan lembaga pemda setempat.
    - Pesan dan saran sarana perbaikan situs web dari saran pengunjung/pengguna situs web seperti forum Diskusi, Saran dan Komentar.
2. Keaktifan situs web yaitu menunjukan kapan konten situs web ditulis dan kapan situs web tersebut diupdate
3. Media Sosial dengan adanya media sosial ini dapat menilai keaktifan pada pemerintahan dalam membantu memberikan informasi pada masyrakat sehingga ini dimasukan dalam pembobotan
    - Facebook
        - Jumlah Update
        - Jumlah Update yang berhubungan dengan pemerintahan
        - Jumlah Fan
    - Twitter



Gambar 2.2: *Roadmap* egovbench

  - Jumlah Update
  - Jumlah Update yang berhubungan dengan pemerintahan
  - Jumlah Tweeet
  - Jumlah Follower
- Youtube
  - Jumlah Update
  - Jumlah Update yang berhubungan dengan pemerintahan
  - Jumlah View
  - Jumlah Subscriber

Egovbench sendiri merupakan sebuah proyek yang kontinyu dan selalu berkembang agar dapat terus meningkatkan kualitas perangkingan yang merupakan core business dari egovbench. Selain itu egovbench juga merencanakan untuk menambah berberapa fitur-fitur pendukung dari fitur perangkingan sebagai *added value*

sehingga perangkingan situs web resmi pemerintahan daerah di Indonesia sendiri tidak stagnan dan dapat dilihat dari berberapa sudut pandang. Pada gambar 2.2 adalah *roadmap* untuk pengembangan egovbench dalam berberapa tahun kedepan.

Pada tahun 2016, egovbech mulai dikembangkan dengan berfokus pada 2 fitur utama yaitu *crawling data* dan perangkingan situs web resmi pemerintah daerah di Indonesia. Pada tahun 2017 ini pengembangan egovbench berfokus kepada fitur *content detection* dan *storage*. *Content detection* sendiri adalah fokus dari penelitian ini sebagaimana yang telah dijelaskan pada bagian sebelumnya. Selain sudah termasuk kedalam roadmap, permasalahan yang muncul pada *web crawling* untuk *crawling data* sendiri juga menegaskan pentingnya *content detection ini*.

Untuk saat ini proses alur aplikasi egovbench dapat dilihat seperti pada gambar 2.3. proses pertama yaitu proses *crawling* untuk setiap link yang ada pada sebuah situs web resmi pemerintah daerah. Proses *crawling* ini dilakukan, selain untuk mencari semua link yang ada pada sebuah situs web, juga untuk menyaring link mana saja yang valid atau dapat diakses dan mengkategoriasasikan setiap link dengan tujuh kategori informasi yang harus ada sesuai dengan Panduan Penyelenggaraan Situs Pemerintah Daerah.

Proses kedua yaitu proses untuk mengambil *main content* dari halaman web yang telah dilakukan *crawling* sebelumnya. proses ini bertujuan untuk mengambil *main content* dari sebuah halaman web agar dapat dilakukan penilaian. Proses pengambilan *main content* dari sebuah halaman web sendiri terbukti merupakan salah satu tantangan yang besar dimana sebuah situs web suatu pemerintah daerah memiliki standar struktur yang berbeda satu dengan yang lain sehingga menyulitkan untuk secara tepat mendapatkan *main content* untuk masing-masing pemerintah daerah. Hal inilah yang menjadi fokus pada penelitian ini.

Proses terakhir yaitu proses penilaian situs web resmi pemerintah daerah sesuai dengan kriteria penilaian pada Panduan Penyelenggaraan Situs Pemerintah Daerah yang telah dipaparkan sebelumnya. Proses penilaian ini sangat bergantung dengan pengambilan main content yang menjadi fokus penelitian ini agar dapat memberikan penilaian yang akurat.

Gambar 2.3: Proses Penilaian Situs Web Resmi Pemerintah Daerah oleh egovbench saat ini

Untuk tahun 2018 direncanakan dilakukan penambahan fitur *data integration* dan *text summarization*. untuk tahun 2019 direncanakan penambahan fitur *Responsive of Social Media*. Ketiga fitur tersebut merupakan fitur yang berfungsi sebagai *added value*, dimana dengan adanya fitur-fitur tersebut diharapkan egovbench sendiri tidak hanya sebagai alat perangkingan namun juga sebagai sudut pandang dan penghubung untuk menilai kualitas situs web resmi pemerintah daerah di Indonesia.

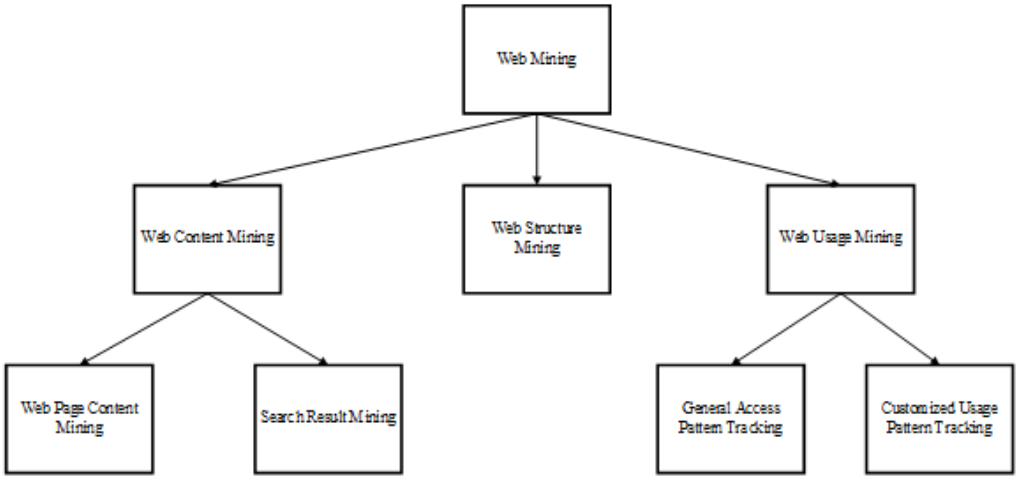
2.1.3 *Web Mining*

Etzioni mendefinisikan *web mining* sebagai teknik dari *data mining* yang bertujuan untuk mengekstrak informasi dari halaman dan layanan web [Etzioni and Oren, 1996]. Definisi lain dari Kumar adalah penggunaan teknik *data mining* untuk informasi yang tidak terstruktur atau semi terorganisir dan secara alami menemukan dan mengekstrak data dan pembelajaran yang bermanfaat dan sebelumnya tidak jelas dari web [Kumar, 2015]. Borges dan Lavene [Borges and Levene, 2000] menjelaskan bahwa secara umum *web mining* secara umum dapat diklasifikasikan menjadi 3 kelompok yaitu (1) *Web Content* Mining (2) *Web Structure Mining* (3) *Web Usage Mining* seperti yang terlihat pada gambar 2.4

Pada penelitian ini akan lebih mengarah kepada *web content mining* terutama *web page content mining* yang disebabkan karena pada penelitian ini berfokus kepada pencarian *main content* pada sebuah halaman web.

2.1.3.1 *Web Content Mining*

Web Content Mining adalah salah satu bagian dari *web mining*. *Web Content mining* mengacu pada penemuan informasi bermanfaat dari konten pada halaman web seperti teks, gambar video dan lain-lain [Kosala and Blockeel, 2000, Prakasam and Suresh, 2010] . Kosala juga menambahkan bahwa sebagian besar dari konten pada



Gambar 2.4: *Web Mining Taxonomy* [TasnimSiddiqui and Aljahdali, 2013]

halaman web adalah *unstructured data content* yang dapat direpresentasikan sebagai sekumpulan kata atau teks. Untuk melakukan proses pengambilan atau ekstraksi konten-konten tersebut dikembangkanlah berbagai macam teknik *web content extraction*.

Web content extractor sendiri adalah alat atau *tool* yang digunakan untuk mengambil atau mengekstrak konten yang ada pada sebuah halaman web. *Web content extraction* sendiri memiliki banyak sekali pendekatan diantaranya yang cukup umum digunakan adalah teknik *machine learning*, teknik visual dan teknik *template*.

(a) Machine learning Pendekatan *machine learning* dalam menentukan *main content* secara umum menggunakan *wrapper* dalam mengolah halaman web untuk dilakukan pendekatan machine learning.*Wrappe* adalah prosedur (program) untuk mengekstrak *database record* dari sumber informasi tertentu, khususnya dari halaman web [Yunis, 2016]. Sedangkan Zeleny mengatakan bahwa *wrapper* adalah program yang nantinya bisa digunakan untuk mengekstrak area tertentu dari halaman web yang diberikan [Zeleny et al., 2017], Serta setiap *wrapper* dapat dianggap sebagai pendeskripsi dari segmen tertentu pada halaman. Liu [Liu, 2011] mengatakan bahwa terdapat 3 cara untuk membangun sebuah *wrapper* yaitu:

(a) *Manual coding Wrapper* yang dapat dibuat oleh seseorang yang akrab dengan *markup* dari halaman web yang berisi data.

(b) *Wrapper Induction* yang dibuat dengan *supervised machine learning* digunakan untuk mendapatkan aturan dalam melakukan ekstraksi. Ini memerlukan kumpulan halaman web dengan data yang relevan yang kemudian diberi label secara manual pada setiap halaman web.

(c) *Automated data extraction* yang dibuat dengan *Unsupervised machine learning* digunakan sebagai pengganti *supervised learning* untuk mendapatkan aturan dalam melakukan ekstraksi. Pada teknik ini tidak membutuhkan untuk memberi label data secara manual di halaman web.

Salah satu penelitian yang menggunakan pendekatan ini adalah penelitian Kohlschütter yang mana menggabungkan pendekatan *wrapper* dengan pendekatan *machine learning* untuk melakukan pencarian *main content* [Kohlschütter et al., 2010]. Dengan menggunakan pendekatan *wrapper*, Kohlschütter membagi halaman web menjadi sebuah blok-blok yang kemudian pada setiap blok tersebut dilakukan pendekatan *machine learning* untuk menentukan apakah blok tersebut merupakan *main content* atau bukan *main content*.

Yao juga menggunakan definisi dan teknik yang digunakan oleh Kohlschütter et al. untuk membagi halaman web menjadi blok untuk kemudian dilakukan pendekatan *machine learning* untuk menentukan *main content* [Yao and Zuo, 2013]. Lundgren mengajukan penggunaan *Random Class Classifier* untuk meningkatkan akurasi dari algoritma *Boilerpaper Library* yang digagas oleh Kohlschütter [Lundgren et al., 2015]. *Random forests* berkerja dengan membuat berberapa *rule decision tree*, biasanya mencapai 100, seperti pada pembuatan *decision tree* seperti pada umumnya namun menggunakan *random variance* sehingga masing-masing sedikit berbeda satu sama lain. Hampir sama dengan pemikiran Kohlschütter yang mana mana blok *main content* seharusnya lebih "padat" dibandingkan blok yang bukan *main content*, Weninger [Weninger et al., 2010] menggunakan pendekatan lain yaitu dengan

merumuskan istilah *Tag Ratio*. *Tag Ratio* sendiri adalah rasio dari jumlah karakter *non*-HTML-*tag* dibandingkan jumlah karakter HTML-*tag* per baris. Jika jumlah karakter *HTML-tag* pada baris tertentu adalah 0 maka rasio diset menjadi jumlah baris. Kemudian dari rasio-rasio tersebut dibuat *Tag Ratio Histogram*. Berdasarkan hasil dari *Tag Ratio Histogram*, jika terdapat baris yang memiliki nilai *Tag Ratio* yang lebih tinggi dibandingkan baris lain maka kemungkinan besar baris tersebut adalah *main content*.

(b) Visual

Pendekatan berbasis visual berfokus kepada analisis fitur visual dari isi dokumen seperti yang dirasakan oleh pembaca manusia [Zeleny et al., 2017]. Pada dokumen HTML, mendapatkan informasi visual yang diperlukan memerlukan pemrosesan dokumen oleh HTML *rendering engine* untuk menghasilkan *style* dan tata letak elemen individual. Penggunaan informasi visual memungkinkan untuk mencapai akurasi segmentasi yang lebih tinggi dibandingkan dengan pendekatan berbasis DOM. Di sisi lain, perlunya *rendering* halaman dan model dokumen yang lebih kompleks yang diproses biasanya dalam banyak tahap membuat pendekatan berbasis visual lebih lambat secara signifikan dan tidak *scalable*.

Cai memperkenalkan teknik *Vision-Based Page Segmentation Algorithm* (VIPS) untuk mendeteksi *main content* yang ada pada halaman web [Cai et al., 2003]. VIPS mencoba untuk mensimulasikan pendekatan berdasarkan pengguna manusia untuk memahami struktur konten dari sebuah halaman web. Manusia tidak melihat *markup* HTML atau DOM pada halaman web, Sebaliknya, manusia hanya melihat tampilan visual dari sebuah halaman web. Oleh karena itu, VIPS mencoba memanfaatkan isyarat spasial dan visual yang sama yang memberi petunjuk pada pengguna manusia tentang struktur konten pada halaman web.

Chen mengajukan salah satu teknik mendeteksi struktur dari web dengan menggunakan pendekatan visual [Chen et al., 2003]. Chen membagi halaman web menjadi bagian-bagian berdasarkan visual yang dinamakan *Explicit*

Separator Detection. *Explicit Separator Detection* membagi halaman web menjadi berberapa partisi atau bagian. Contohnya Chen mendefinisikan bahwa *left side bar* adalah satu perempat bagian dari halaman web pada bagian kiri dan *right side bar* adalah satu perempat bagian dari halaman web pada bagian kanan.

Baluja dan Shumeet menggunakan pendekatan *entropy reduction* dalam melakukan penentuan *main content* [Baluja and Shumeet, 2006]. Pada penelitian tersebut, halaman web dirubah kedalam bentuk DOM yang kemudian akan dilakukan pembagian menjadi segmen-segmen dengan *decision tree* yang dibuat oleh peneliti dan dibantu dengan tampilan visual.

Garis pemutus yang digunakan untuk membagi halaman web menjadi segmen-segmen dipilih berdasarkan *Information Gain* yang muncul pada setiap garis. Selain itu pemutusan menjadi segmen-segmen tersebut juga dibantu dengan *Entropy Reduction* untuk menimalisir *noise* yang muncul.

(c) *Template*

Template dapat didefinisikan sebagai tata letak halaman web dengan tempat atau bagian dimana isi variabel dapat dimasukkan [Yunis, 2016]. Contohnya untuk halaman deskripsi produk pada situs e-commerce tertentu biasanya memiliki tata letak visual yang sama, hal ini berlaku juga untuk halaman web pada situs web resmi pemerintahan daerah di Indonesia. Sedangkan konten-konten lainnya yang muncul berulang kali seperti menu navigasi umumnya disebut dengan *template-generated content* [Gottron., 2009]. Hal ini juga sering disebut dengan *boilerplate detection*.

Template detection bertujuan untuk mengidentifikasi dan menyaring bagian-bagian dari sebuah halaman web yang berulang kali muncul pada halaman yang mirip [Zeleny et al., 2017]. *Template detection* sendiri diasumsikan bahwa *main content* atau konten yang relevan akan tetap ada meskipun bagian-bagian tersebut dihilangkan [Alarte et al., 2015, Barua et al., 2014, Gao and Fan, 2014]

Lin dan Ho [Lin and Ho, 2002] mendefinisikan *webpage cluster* sebagai satu

set halaman web yang didasarkan pada *template* yang sama. Sehingga untuk dapat mengenali *structure template* dari sebuah halaman web diperlukan *training set* dari halaman web yang berasal dari *webpage cluster* yang sama. Dengan kata lain untuk mendapatkan *template* pada halaman web pada suatu situs web diperlukan halaman web lain yang ada pada situs web tersebut.

2.1.4 *Document Object Model*

World Wide Web Consortium mendefinisikan *Document Object Model* (DOM) adalah API pemrograman untuk dokumen *HyperText Markup Language* (HTML) dan *eXtensible Markup Language* (XML) [W3C, 2017]. Dalam spesifikasi DOM, istilah dokumen dipergunakan secara luas, XML digunakan sebagai cara untuk mewakili berbagai jenis informasi yang mungkin tersimpan dalam beragam sistem, dan sebagian besar dulunya dilihat sebagai data bukan sebagai dokumen. Meskipun demikian, XML menyajikan data tersebut sebagai dokumen, dan DOM dapat digunakan untuk mengelola data tersebut.

Dengan *Document Object Model*, kita dapat membuat dokumen, menavigasi struktur mereka, dan menambahkan, memodifikasi, atau menghapus elemen dan konten. Apa pun yang ditemukan dalam dokumen HTML atau XML dapat diakses, diubah, dihapus, atau ditambahkan menggunakan *Document Object Model*, dengan beberapa pengecualian - khususnya untuk antar muka DOM untuk *subset internal* dan *subset eksternal* yang belum ditentukan.

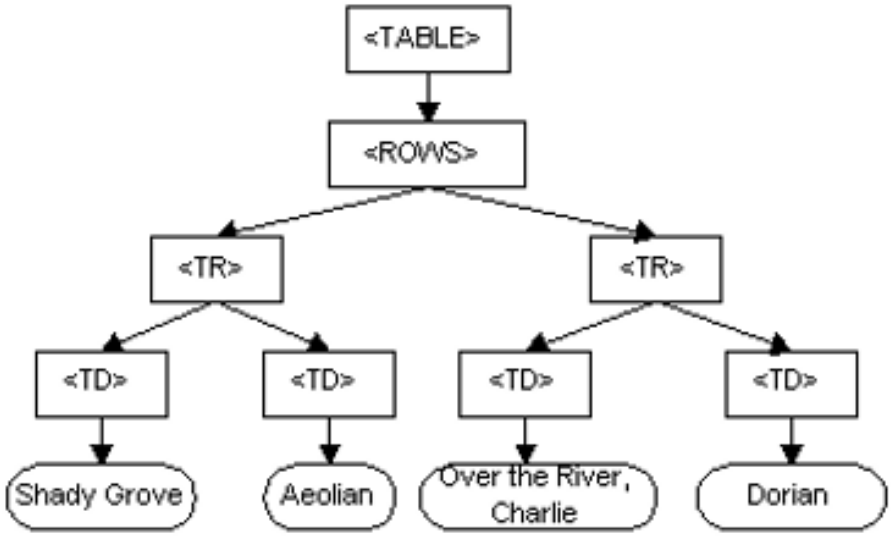
Sesuai dengan spesifikasi yang telah ditentukan oleh W3C, salah satu tujuan penting untuk *Document Object Model* adalah menyediakan antarmuka pemrograman standar yang dapat digunakan di berbagai lingkungan dan aplikasi. *Document Object Model* dapat digunakan dengan bahasa pemrograman apapun. Pada kode 2.1 berikut ini adalah contoh tabel pada sebuah dokumen HTML.

```
1  <TABLE>
2      <ROWS>
3          <TR>
4              <TD>SHADY GROVE</TD>
5              <TD>AEOLIAN</TD>
6          </TR>
7          <TR>
8              <TD>OVER THE RIVER, CHARLIE</TD>
```

```
9                   <TD>DORIAN</TD>
10                </TR>
11             </ROWS>
12          </TABLE>
```

Kode 2.1: Contoh *tag* HTML untuk sebuah tabel

Dari kode tabel HTML seperti yang terlihat pada kode 2.1, maka representasi dari *Document Object Model* untuk tabel tersebut adalah seperti pada gambar 2.5.



Gambar 2.5: representasi DOM untuk tabel pada kode 2.1

Dalam *Document Object Model*, dokumen memiliki struktur logis yang mirip dengan “pohon”, Untuk lebih tepatnya, lebih dapat dikatakan ”hutan” yang bisa menampung lebih dari satu pohon. Namun, *Document Object Model* tidak menentukan bahwa dokumen diimplementasikan sebagai pohon, juga tidak menentukan bagaimana hubungan antar objek dapat dilaksanakan dengan cara apapun.

Dengan kata lain, model objek menentukan model logis untuk antarmuka pemrograman, dan model logis ini dapat diimplementasikan dengan cara apa pun sehingga implementasi tertentu dapat ditemukan dengan mudah. Dalam spesifikasi ini, W3C menggunakan istilah model struktur untuk menggambarkan representasi mirip pohon dari sebuah dokumen, W3C secara khusus menghindari istilah seperti ”pohon” atau ”hutan” agar tidak menyiratkan penerapan tertentu. Salah satu ciri penting model struktur DOM adalah isomorfisma structural dimana jika ada dua

implementasi *Document Object Model* yang digunakan untuk membuat representasi dokumen yang sama, mereka akan membuat model struktur yang sama, dengan objek dan hubungan yang sama persis.

Nama ”*Document Object Model*” dipilih karena merupakan ”model objek” yang digunakan dalam pengertian desain berorientasi objek tradisional: dokumen dimodelkan menggunakan objek, dan model tidak hanya mencakup struktur dokumen, namun juga perilaku sebuah dokumen dan benda-benda yang disusunnya. Dengan kata lain, simpul pada diagram di atas tidak mewakili struktur data, mereka mewakili objek, yang memiliki fungsi dan identitas. Sebagai model objek, *Document Object Model* mengidentifikasi:

1. Antarmuka dan objek yang digunakan untuk mewakili dan memanipulasi dokumen
2. Semantik dari antarmuka dan objek ini termasuk perilaku dan atribut
3. Hubungan dan kolaborasi di antara antarmuka dan objek ini

Struktur dokumen SGML secara tradisional diwakili oleh model data abstrak, bukan oleh model objek. Dalam model data abstrak, model berpusat di sekitar data. Dalam bahasa pemrograman berorientasi objek, data itu sendiri dienkapsulasi pada objek yang menyembunyikan data, melindunginya dari manipulasi eksternal secara langsung. Fungsi yang terkait dengan objek ini menentukan bagaimana objek dapat dimanipulasi, dan itu adalah bagian dari model objek.

Document Object Model saat ini terdiri dari dua bagian, DOM *Core* dan DOM HTML. DOM Core mewakili fungsionalitas yang digunakan untuk dokumen XML, dan juga berfungsi sebagai dasar untuk DOM HTML. Semua implementasi DOM harus mendukung antarmuka yang terdaftar sebagai ”fundamental” dalam spesifikasi Core. Selain itu, implementasi XML harus mendukung antarmuka yang terdaftar sebagai *”extended”* dalam spesifikasi Core. Spesifikasi HTML DOM Level 1 mendefinisikan fungsionalitas tambahan yang dibutuhkan untuk dokumen HTML.

Document Object Model berasal sebagai spesifikasi untuk memungkinkan skrip JavaScript dan program Java menjadi portabel di antara browser web. *Dynamic* HTML adalah leluhur langsung dari *Document Object Model*, dan pada awalnya di-

pikirkan sebagian besar dari segi browser. Namun, ketika Document Object Model Working Group dibentuk, Document Object Model Working Group juga bergabung dengan vendor di domain lain, termasuk editor HTML atau XML dan repositori dokumen.

Dalam antarmuka DOM secara mendasar, tidak ada objek yang mewakili entitas. Referensi karakter numerik, dan referensi ke entitas yang telah ditentukan sebelumnya dalam HTML dan XML, digantikan oleh karakter tunggal yang membentuk penggantian entitas. Misalnya pada contoh kode 2.2.

```
1    <p> This is a dog &amp; a cat </p>
```

Kode 2.2: contoh sebuah *tag* HTML

”& amp;” akan diganti dengan karakter ”&”, dan teks dalam elemen <p> akan membentuk urutan karakter tunggal yang kontinyu. Representasi entitas umum, baik internal maupun eksternal, didefinisikan dalam antarmuka (spesifikasi) perluasan dari spesifikasi Tingkat 1.

DOM menentukan antarmuka yang dapat digunakan untuk mengelola dokumen XML atau HTML. Penting untuk disadari bahwa antarmuka ini adalah abstraksi - sama seperti ”kelas dasar abstrak” di C++, ini adalah alat untuk menentukan cara untuk mengakses dan memanipulasi representasi internal aplikasi sebuah dokumen. Secara khusus, interface tidak menyiratkan implementasi konkret tertentu. Setiap aplikasi DOM bebas untuk memelihara dokumen dalam representasi yang mudah digunakan, asalkan antarmuka yang ditunjukkan dalam spesifikasi ini didukung. Beberapa implementasi DOM akan menjadi program yang ada yang menggunakan antarmuka DOM untuk mengakses perangkat lunak yang ditulis jauh sebelum spesifikasi DOM ada. Oleh karena itu, DOM dirancang untuk menghindari ketergantungan implementasi. Sejauh ini Document Object Model telah memiliki berberapa iterasi yaitu:

1. DOM Level 1 menyediakan model lengkap untuk keseluruhan dokumen HTML atau XML, termasuk sarana untuk mengubah bagian dokumen apa pun.
2. DOM Level 2 diterbitkan pada akhir tahun 2000. Ini mengenalkan fungsi

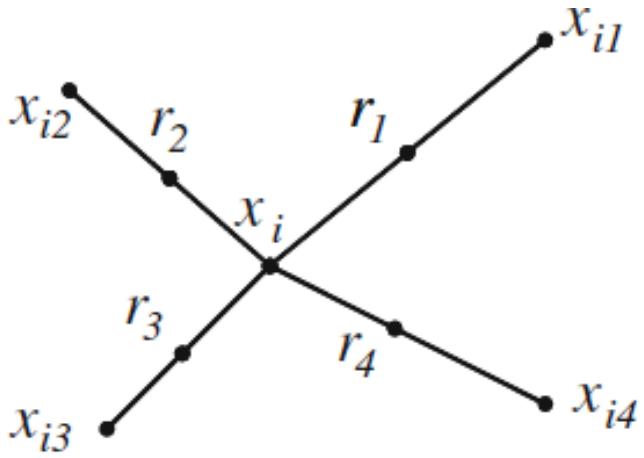
getElementById serta model acara dan dukungan untuk namespace XML dan CSS.

3. DOM Level 3, yang diterbitkan pada bulan April 2004, menambahkan dukungan untuk penanganan *event* XPath dan keyboard, serta sebuah antarmuka untuk mengumpulkan dokumen sebagai XML.
4. DOM Level 4 diterbitkan pada tahun 2015. Ini adalah cuplikan dari standar yang dimiliki oleh Web Hypertext Application Technology Working Group (WHATWG).

2.1.5 SMOTE-ENN

Synthetic Minority Over-sampling Technique (SMOTE) yang dikembangkan oleh [Chawla et al., 2002] merupakan salah satu metode dalam melakukan balancing dataset dengan menggunakan pendekatan *oversampling* dengan melakukan oversampling pada setiap label atau class minoritas dengan membuat data sintetis pada setiap segmen baris yang menghubungkan setiap atau seluruh kelas atau label pada *k-nearest neighbors*. Dengan melihat jumlah *oversampling* yang diperlukan, *neighbors k-nearest neighbors*. Implementasi dari pendekatan ini menggunakan *euclidean distance* untuk melakukan balancing pada setiap kelas atau label sehingga mendekati distribusi 50% atau seimbang antara masing-masing kelas atau label. SMOTE bertujuan untuk membentuk contoh atau data pada kelas minoritas dengan melakukan interpolasi diantara berbera contoh atau data pada kelas minoritas dimana permasalahan *overfitting* dapat dihindari dan *decision boundaries* pada kelas minoritas menyebar lebih jauh pada kelas mayoritas [Luengo et al., 2011].

Akan tetapi *cluster* pada kelas atau label dapat tidak terdefinisi dengan baik dikarenakan kelas atau label minoritas dapat mengganggu kelas mayoritas ketika oversampling dilakukan dimana hal ini dapat menyebabkan *overfitting*. Batista mengajukan variasi dari SMOTE dengan menggabungkan dengan *Edited Nearest Neighbors* yang dinamakan SMOTE-EEN untuk mem-*filter* data atau contoh yang terbentuk dari over-sampling yang dilakukan dari SMOTE dengan menghilangkan contoh atau data yang terbentuk yang memiliki kelas atau label berbeda dibandingk-



Gambar 2.6: Contoh SMOTE meng-interpolasi data baru untuk melakukan *over sampling* [Luengo et al., 2011]

an dua dari tiga contoh atau data terdekat (*nearest neighbors*) [Batista et al., 2004].

2.2 Kajian Penelitian Terdahulu

Pada bagian ini akan dijelaskan beberapa penelitian yang terkait dengan penelitian yang akan dilakukan.

2.2.1 *Largest Pagelet*

Teknik *Largest Pagelet* adalah teknik *Template Detection* yang dikembangkan oleh Bar Yossef dan Rajagopalan [Bar-Yossef and Rajagopalan, 2002]. Definisi *Pagelet* sendiri menurut Chakrabarti oleh Yossef adalah bidang logikal yang berdiri sendiri (*self-contained*) di dalam halaman web yang memiliki topik atau fungsionalitas yang terdefinisikan dengan baik. Setiap halaman web dapat diurai atau dipecah kedalam satu atau berberapa *pagelet* sesuai dengan topik atau fungsionalitas yang muncul pada halaman web tersebut. Yossef mengatakan bahwa pagelet lebih baik dibandingkan menggunakan halaman web untuk *Information Retrieval* karena *pagelet* secara struktur lebih kohesif dan lebih selaras dengan T*opical Unity Principle* dan *Relevant Linkage Principle*.

Dalam implementasi pagelet untuk *template detection*, Yossef menggunakan teknik *shingle/shingling* yang dikembangkan oleh Broder [Bar-Yossef and Rajago-

palan, 2002]. *Shingl*e adalah sebuah *text fingerprint* yang tidak berubah (*invariant*) dalam gangguan yang kecil (*small pertubation*). *Pagelet shingle* yang memiliki kemunculan melebihi batas yang ditentukan dijadikan sebagai *template*.

2.2.2 *Template Hash*

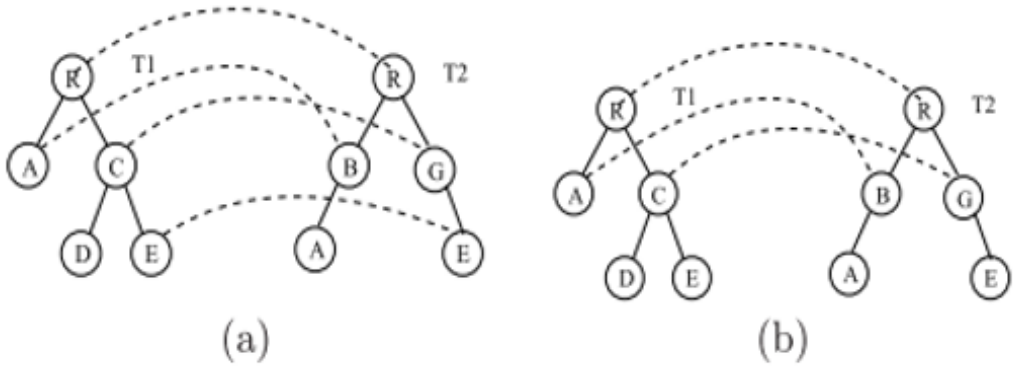
Gibson mengajukan teknik *template detection* dengan melakukan *hashing* pada setiap tag yang ada pada halaman web [Gibson et al., 2005]. Hasil dari hash untuk setiap tag dinamakan *template hash*. Lalu dengan menghitung frekuensi kemunculan setiap *template hash* yang terdapat pada setiap halaman web pada situs web maka dapat menggambarkan seberapa sering HTML *node* tersebut terlihat atau berada pada halaman web.

Dari informasi tersebut maka dibangun *template node* pada setiap halaman dimana *template node* harus memenuhi 2 syarat yaitu (1) Jumlah kemunculan node dari *template hash* harus berada dalam batas yang ditentukan (2) *node template hash* tersebut bukan *child* dari *node template hash* yang lain. Kemudian *template node* digabungkan menjadi satu untuk membuat *template* dari halaman web.

2.2.3 *Content Extractor*

Content Extractor adalah teknik yang digunakan oleh Debnath et al. untuk melakukan *template detection* dengan menggunakan pendekatan *Inverse Block Document Frequency* [Debnath et al., 2005]. Halaman web dibagi menjadi berberapa blok tertentu. Pembagian halaman web menjadi blok ini mengunakan aturan yang dibuat oleh peneliti berdasarkan hasil observasi yang mereka lakukan dengan melihat bagaimana pola-pola umum dari *tag-tag* html pada situs web yang ada. Kemudian setelah itu konten dari setiap blok yang ada dilakukan *Inverse Block Document Frequency*.

Jika suatu blok muncul pada berberapa halaman web maka nilai dari *Inverse Block Document Frequency* akan lebih kecil daripada blok yang hanya muncul pada satu halaman web saja. Blok dengan nilai kecil tersebut dapat disebut sebagai *template* dari halaman web dan blok yang jarang muncul atau muncul hanya sekali



Gambar 2.7: Contoh *top down mapping normal* (a) dan *restricted* (b) [Vieira et al., 2006]

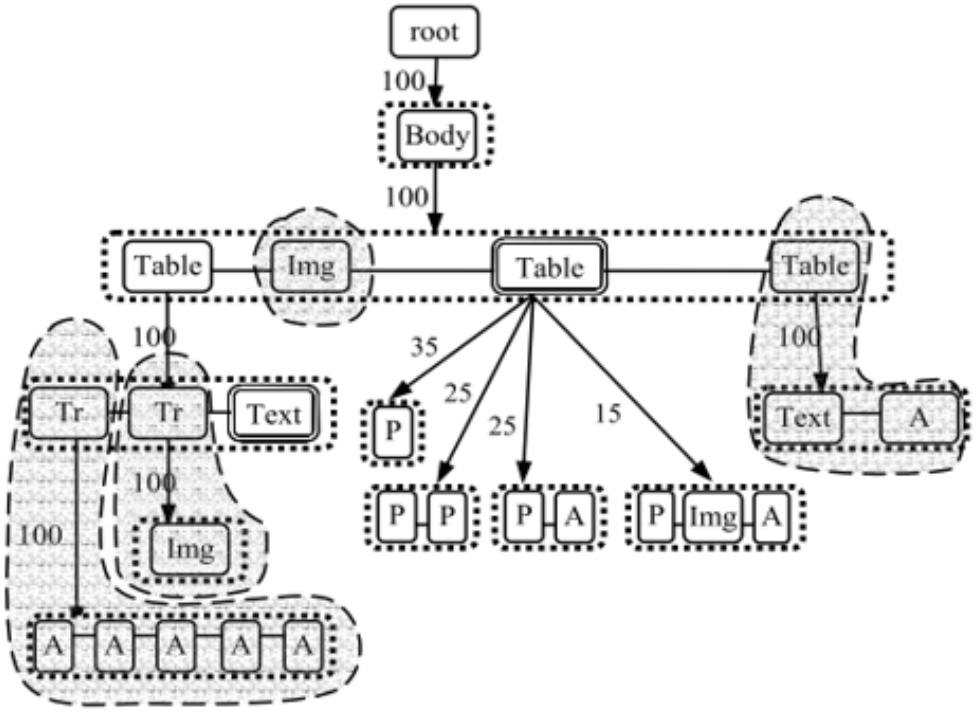
saja pada halaman web dapat dikatakan sebagai *main content* halaman web.

Content Extractor sendiri merupakan bagian dari *Feature Extraction* yang dikembangkan juga oleh Debnath. dimana jika *Content Extractor* memerlukan kumpulan halaman web, maka *Feature Extractor* hanya memerlukan fitur set dan fitur yang dipilih untuk menentukan *main content* pada sebuah halaman web.

2.2.4 RTDM-TD

Vieira dalam melakukan *template detection* menggunakan RTDM-TD yang dikembangkan dari *Restricted Top-Down Mapping* [Vieira et al., 2006]. Halaman web direpresentasikan sebagai *labeled ordered rooted tree* sesuai dengan DOM *tree*. Secara umum pendekatan *Restricted Top-Down Mapping* membatasi operasi *remove* dan *insert* yang ada pada *top-down mapping* dengan hanya beroperasi sebatas pada *leaves* pada struktur *tree* seperti yang terlihat pada gambar 2.7.

RTDM-TD tidak memerlukan *input threshold*, dimana berbeeda dengan proses *Restricted Top-Down Mapping* yang akan berhenti jika *partial cost* sudah melebihi batas, RTDM-TD akan menyelesaikan operasi tanpa menghiraukan batas *partial cost*. Selain itu RTDM-TD mengingat mengenai kejadian-kejadian dimana tidak ada operasi *insertion, removal* atau *update* pada*node*. Setelah *common subtree* ditemukan maka *common subtree* tersebut direpresentasikan sebagai *template* halaman web.



Gambar 2.8: Contoh dari *Site Style Tree* [Yi et al., 2003]

2.2.5 *Site Style Tree*

Yi et al. melakukan pendekatan yang unik dimana dalam penelitiannya Yi menggunakan *Style Tree* dalam melakukan *template detection* [Yi et al., 2003]. *Style Tree* ini dibangun berdasarkan hasil observasi dari Yi dalam melihat situs web secara umum seperti yang terlihat pada gambar 2.8.

S*tyle Tree* digunakan untuk menemukan *style* umum di halaman web pada situs web untuk menghilangkan *noise*. *Style Tree* dapat mengkompress *style* dari tampilan umum dari kumpulan halaman web. Dari *Style Tree* dibangun *Style Node* dimana *Style Node* merupakan urutan dari tag node pada DOM *tree*. Dari *Style Node* ini kemudian dibangun *Site Style Tree*.

2.2.6 TeMex

Alarte mengembangkan sebuah *tool* untuk melakukan ekstraksi *template* yang dinamakan *Template Extractor* (TeMex) [Alarte et al., 2015] Untuk melakukan *template detection* dari sebuah halaman web, TeMex mencari terlebih dahulu kandidat ha-

laman web yang memiliki kesamaan *template* dengan halaman web yang ingin dicari templatenya. Pencarian kandidat ini menggunakan teknik *complete sub-digraph*.

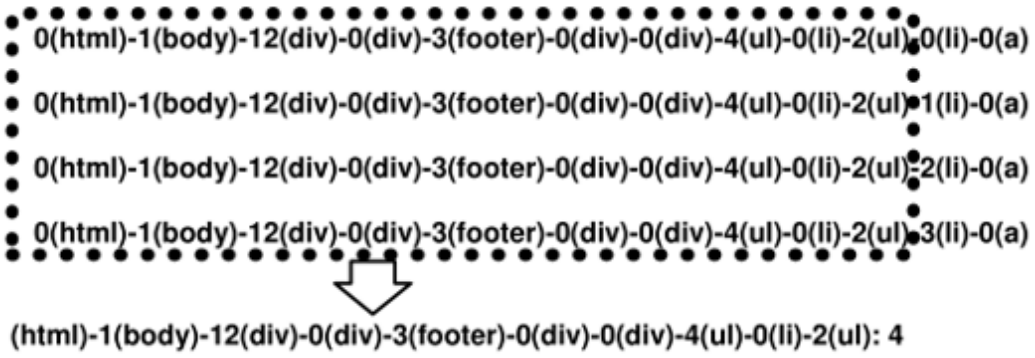
Kandidat terbaik kemudian dijadikan satu dengan halaman web yang ingin dicari *template*-nya menjadi sebuah kumpulan halaman web. Kemudian dari kumpulan halaman web tersebut dilakukan pembuatan *key page*. *Key page* dibuat dengan melakukan komparasi struktur DOM dengan pendekatan *Equal Top-Down Mapping*. Dari *Key Page* yang telah dilakukan komparasi tersebut dibentuk *template* dari halaman web.

2.2.7 StaDyNoT

Barua mengembangkan sebuah *tool* untuk melakukan ekstraksi artikel berita yang dari halaman web dinamakan *Static Noise Tags and Dynamic Noise Tags* (StaDyNoT) [Barua et al., 2014]. Untuk mencari *main content* pada halaman web yang ingin dicari, Pada teknik StaDyNoT dilakukan pencarian terlebih dahulu mengenai *neighbor article web pages* yaitu set halaman web yang berada dalam satu situs web dan memiliki kategori yang sama dengan halaman web yang ingin dilakukan pencarian *main content*.

Setelah *neighbor article web pages* ditemukan kemudian halaman web yang ingin dicari dan *neighbor article web pages* dirubah kedalam bentuk DOM. Kemudian dilakukan pencarian *Static Noise Tag*. *Static Noise Tag* dicari dengan membentuk *key* yang berisi *tag name*, *tag attribute* dan *tag data* yang kemudian dicari frekuensi kemunculan untuk setiap *key* pada seluruh halaman web tersebut. Jika *key* memiliki frekuensi kemunculan yang sama dengan jumlah halaman web pada *neighbor article web pages* maka key tersebut ditandai sebagai *Static Noise Tag*.

Setelah dilakukan pencarian *Static Noise Tag*, kemudian dilakukan pencarian *Dynamic Noise Tags*. *Dynamic Noise Tags* ditemukan dengan menggunakan pendekatan *Least Common Ancestor (LCA)* seperti yang terlihat pada gambar 2.9. Dengan menggunakan DOM tree, untuk setiap node dengan *tag* “a” direpresentasikan dengan *path-string*. *Path string* dari *node n* adalah *path* dari *root* menuju *node n* pada DOM *tree* dengan informasi posisi dari setiap *node* di *path* yang ada.



Gambar 2.9: Ilustrasi identifikasi *Dynamic Noise Tags* dengan menggunakan pendekatan *Least Common Ancestor* (LCA) [Barua et al., 2014]

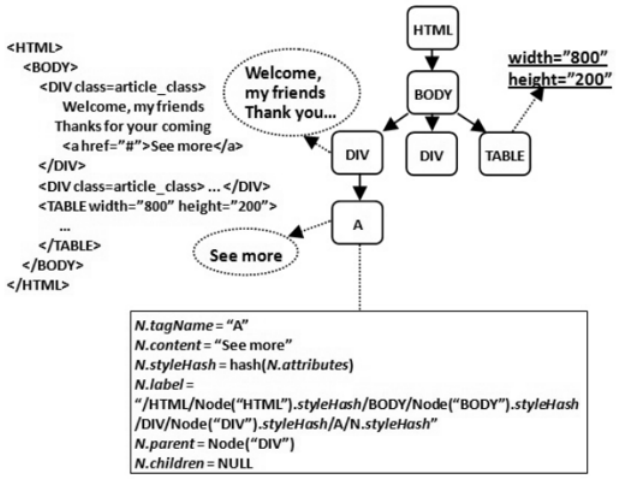
Setelah ditemukan *Static Noise Tags and Dynamic Noise Tags*, Kemudian dengan menghilangkan *node* yang di tandai *noise* pada proses *Static Noise Tags* dan *Dynamic Noise Tags* dan menghilangkan *non-informative Formatting Tags* seperti *script* maka *main content* dapat ditemukan. Proses penandaan *node* menjadi *noise* ini dapat kita asumsikan sebagai pembuatan *template* pada halaman web.

2.2.8 *Site-oriented Segment Object Model*

Gao dan Fan mengembangkan *Site-oriented Segment Object Model* dalam mencoba mencari *template* dari halaman web [Gao and Fan, 2014]. *Site-oriented Segment Object Model* adalah deretan atau jajaran dari DOM tree pada halaman di situs web. Gao mengemukakan bahwa umumnya *template* muncul dalam bentuk segmen contohnya seperti segmen navigasi dan segmen *main content*, Sehingga untuk penelitiannya, Gao menggunakan segmen sebagai granularitas dari *template detection*.

Halaman web umumnya memiliki konten informasi dan konten yang redundan. Halaman web yang telah diubah menjadi DOM *tree* kemudian dijadikan sebagai *Segment Object Model* (SOM) seperti yang terlihat pada gambar

Segment Object Model Tree dapat merepresentasikan *content* dan *style* dari sebuah halaman web namun tidak dapat merepresentasikan keseluruhan halaman web pada situs web. Untuk mengatasi hal tersebut *Site-oriented Segment Object Model* dibuat dengan merangkum keseluruhan *Segment Object Model Tree* pada



Gambar 2.10: Contoh SOM *Tree* [Gao and Fan, 2014]

setiap halaman web pada situs web.

Setelah Membuat *Site-oriented Segment Object Model*, Gao kemudian melakukan *shingle* dan *cluster segment* untuk membuat *classifier* yang digunakan untuk membuat *template* situs web.

2.2.9 *Schema Inference*

Dalam melakukan *template detection*, Krishna dan Dattatraya melakukan berberapa lanngkah [Krishna and Dattatraya, 2015]. Langkah pertama yaitu membuat DOM *tree* dari halaman web masukan, yang selanjutnya untuk setiap halaman web dilakukan segmentasi dengan menggunakan pendekatan *Vision-based Page Segmentation (VIPS)* untuk membentuk VB *tree*. Selanjutnya dilakukan komparasi blok di VB *tree* yang mana hasilnya akan terdeteksi *Fixed* atau *Variant template* di halaman web. Langkah selanjutnya dilakukan penghapusan *noise block*. *Noise block* adalah blok pada halaman web yang berisi data seperti iklan atau navigasi. Penghapusan *Noise block* dilakukan dengan melakukan perhitugan persentasi area pada setiap node. Setelah dilakukan penghapusan *Noise block* kemudian dilakukan penggabungan *tree* untuk dijadikan *template*. Penggabungan *tree* dilakukan dengan berberapa langkah yaitu (1) *Identification of peer nodes* (2) *Alignment of matrix* (3) *Repetitive Pattern Mining* (4) *Merging of optional nodes*.

2.2.10 *Dice-coefficient*

Kulkarni mengajukan *Dice-coefficient* dalam melakukan *template detection* pada halaman web [Kulkarni et al., 2015]. Konsep utama dari *Dice-coefficient* adalah mencari kemiripan kata atau *string* diantara berberapa dokumen dalam hal ini halaman web yang telah dirubah menjadi DOM *tree*.

Pada *Dice-coefficient string* dirubah menjadi bentuk bigram yang kemudian dilakukan pencarian banyaknya string umum diantaranya dan menggunakan *Dice-coefficient* untuk menghitung kemiripan. Nilai dari *Dice-coefficient* adalah berkisar dari 0 sampai 1 dengan nilai 1 adalah mirip sempurna atau sama persis.

2.2.11 *Shallow Text Feature Set*

Penelitian Kohlschütter ini mmenggunakan pendekatan klasifikasi untuk melakukan pencarian *main content* dengan menggunakan *shallow text feature set* untuk menilai *link density*, dan *text density* [Kohlschütter et al., 2010]. Salah satu dasar dari hal ini adalah blok *main content* seharusnya lebih "padat" dibandingkan blok yang bukan *main content*.

Untuk membentuk halaman web menjadi blok-blok, Kohlschütter merubah halaman web kedalam bentuk blok yang dinamakan *atomic text block* yang dianotasikan dengan fiturnya. *Atomic text block* adalah urutan dari karakter data yang dipisahkan oleh satu atau lebih HTML tag kecuali tag "A" yang bertujuan untuk menilai *link density*.

Text Density adalah perhitungan jumlah *token* yang ada pada blok teks tertentu yang dibagi dengan jumlah baris yang tercover setelah dilakukan *text word-wrapping* pada *fixed column width*. Hasil dari fitur set tersebut kemudian digunakan pada *machine learning* untuk menentukan *main content*.

2.2.12 *Tag Ratio*

Hampir sama dengan pemikiran Kohlschütter [Kohlschütter et al., 2010], dimana blok *main content* seharusnya lebih "padat" dibandingkan blok yang bukan *main content*, Weninger et al. menggunakan pendekatan lain yaitu dengan merumusk-

an istilah Tag Ratio untuk melakukan *template detection* [Weninger et al., 2010]. *Tag Ratio* sendiri adalah rasio dari jumlah karakter non-HTML-*tag* dibandingkan jumlah karakter HTML-*tag* per baris. Jika jumlah karakter HTML-*tag* pada baris tertentu adalah 0 maka rasio diset menjadi jumlah baris. Kemudian dari rasio-rasio tersebut dibuat *Tag Ratio Histogram*. Berdasarkan hasil dari *Tag Ratio Histogram*, jika terdapat baris yang memiliki nilai *Tag Ratio* yang lebih tinggi dibandingkan baris lain maka kemungkinan besar baris tersebut adalah *main content*. Selanjutnya *K-means clustering* dilakukan untuk menentukan grup yang berupa *main content* atau bukan *main content*.

2.2.13 *Text Block Machine Learning*

Yao dalam membagi halaman web menjadi blok-blok dan melakukan pendekatan *machine learning* dalam menentukan *main content* [Yao and Zuo, 2013]. Dalam penelitian Yao tersebut, dia membagi halaman web menjadi sebuah blok-blok sesuai dengan definisi yang dipaparkan pada penelitian Kohlschütter [Kohlschütter et al., 2010]. Pembagian menjadi blok-blok tersebut dilakukan dengan menggunakan HTML *parser* yang memilah HTML DOM *tree* pada halaman web tersebut.

Untuk menentukan apakah sebuah blok merupakan *main content* atau bukan *main content*, Yao menggunakan 3 tipe fitur yaitu : *text features*, *relative position*, dan *id&class token feature*. *Text Feature* adalah hasil ekstraksi yang diambil berdasarkan properti teks pada blok yang ada. Text Feature ini terdiri atas 7 poin fitur yaitu:

1. jumlah kata dalam blok atau tag dan hasil bagi dengan blok sebelumnya
2. Panjang kalimat rata-rata di blok atau tag dan hasil bagi dengan blok sebelumnya.
3. kepadatan teks di blok atau tag dan hasil bagi dengan blok sebelumnya.
4. link di blok atau tag.

Relative position adalah posisi relatif dari blok pada halaman web dimana spesifiknya jika sebuah halaman web dibagi menjadi N blok naka Yao mendiskretkan posisi mereka ke posisi relatif M yang sesuai dengan Gambar 2.11.

$$relative\_pos(n) = \lfloor \frac{n}{N} \times M \rfloor$$

Gambar 2.11: Posisi Relatif oleh Yao [Yao and Zuo, 2013]

Id & class token feature digunakan untuk menangkap informasi semantik yang ada pada HTML, misalnya *token* seperti ad dan nav umumnya mengindikasikan bahwa *element* yang terasosiasi dengan token tersebut bukan *main content*. Untuk menentukan blok merupakan *main content* atau bukan *main content*, Yao menggunakan SVM classifier dalam melakukan pendekatan *machine learning* tersebut.

2.2.14 Rangkuman Penelitian Terdahulu

Berikut ini akan disajikan kesimpulan dan perbandingan dari berberapa penelitian terdahulu seperti yang terlihat pada tabel 2.1 dan tabel 2.2

Tabel 2.1: Rangkuman Penelitian Terdahulu (Template-Based)

Penelitian	Teknik	Penjelasan
(Bar-Yossef & Rajagopalan, 2002)	Frekuensi Pagelet	Setiap halaman web dipecah menjadi pagelet dan template dibentuk berdasarkan frekuensi kemunculan pagelet pada kumpulan halaman web
(Gibson et al., 2005)	Frekuensi Template Hash	Setiap tag diubah menjadi bentuk hash dan template dibentuk berdasarkan frekuensi kemunculan hash pada kumpulan halaman web
(Debnath et al., 2005)	Inverse Block Document Frequency	Halaman web dipecah menjadi berberapa blok kemudian setiap blok dilakukan Inverse Block Document Frequency pada sekumpulan halaman web yang ada untuk membentuk template

Tabel 2.1: Rangkuman Penelitian Terdahulu (Template-Based)

Penelitian	Teknik	Penjelasan
(Vieira et al., 2006)	Tree Mapping	Halaman web dirubah menjadi bentuk labeled ordered rooted tree kemudian dilakukan restricted top down mapping TD untuk setiap halaman web yang ada untuk membentuk template
(Alarte et al., 2015)	Komparasi Key Page	Halaman web dirubah menjadi ke dalam bentuk Key Page yang kemudian dibandingkan dengan kumpulan key page yang lain untuk membentuk template
(Barua et al., 2014)	Frekuensi Key dan Least Common Ancestor	Setiap tag pada halaman web dibentuk menjadi sebuah key yang kemudian dilakukan komparasi dengan kumpulan halaman web yang lain. Selain itu juga dilakukan penggunaan Least Common Ancestor untuk menghilangkan noise tag agar dapat dibentuk template.
(Krishna & Dattatraya, 2015)	Frekuensi blok	Halaman web dirubah kedalam bentuk segmen dengan menggunakan VIPS lalu dilakukan komparasi dengan halaman web lai n untuk membentuk template
(Kulkarni et al., 2015)	Frekuensi kemiripan kata	Halaman web dirubah menjadi kedalam bentuk bigram dan dilakukan pencarian kemiripan bigram dengan halaman web lain untuk membentuk template
(Yi et al., 2003)	Page style tree	Halaman web dirubah menjadi ke dalam bentuk page style tree kemudian dibentuk site style tree untuk membentuk template

Tabel 2.1: Rangkuman Penelitian Terdahulu (Template-Based)

Penelitian	Teknik	Penjelasan
(Gao & Fan, 2014)	Site-oriented Segment Object	Template dibangun dengan membentuk Site-oriented Segment Object.

Tabel 2.2: Rangkuman Penelitian Terdahulu(*Machine Learning*)

Penelitian	Classifier	
(Kohlschütter et al., 2010)	Multiple Classfier	Halaman web dirubah ke dalam bentuk blok dan menggunakan menggunakan shallow text feature set untuk menilai link density, dan text density.
(Yao & Zuo, 2013)	SVM	Halaman web dirubah ke dalam bentuk blok dan menggunakan menggunakan 3 tipe fitur yaitu : text features, relative position, dan id&class token feature
(Weninger, Hsu, & Han, 2010)	K-Means Clustering	Halaman web dirubah menjadi kedalam bentuk tag ratio dimana selanjutnya k-means clustering digunakan untuk menentukan bagian mana yang merupakan *main content*

2.3 Kondisi Kekinian Situs Web Resmi Pemerintah Daerah di Indonesia

Dengan total jumlah pemerintah daerah di Indonesia yang berjumlah 548 pemerintah daerah, jumlah pemerintah daerah yang memiliki situs web resmi berjumlah sebanyak 530 pemerintah daerah. Dari 530 pemerintah daerah yang memiliki web resmi pemerintah daerah tersebut, diperkirakan bahwa terdapat 181 pemerintah daerah yang membangun situs web resmi mereka dengan berdasarkan *content management system* seperti wordpress, joomla atau drupal. Sedangkan 349 pemerintah daerah yang lain, membangun situs web mereka dengan membangung dari awal

dimana sebagian besar menggunakan bahasa pemprograman PHP atau *framework* yang berdasarkan PHP seperti Code Igniter.

Dari 530 situs web resmi pemerintah daerah di Indonesia, Didapati bahwa berberapa situs web memiliki permasalahan-permasalahan yang dapat mengganggu aksesibilitas dan kegunaan situs web itu sendiri. Permasalahan-permasalahan tersebut meliputi :

1. Situs web yang belum atau tidak fungsional seperti situs web yang bermasalahan dengan pihak luar (contohnya permasalahan mengenai *account suspended* dari pihak hosting dan permasalahan situs web yang diretas oleh pihak luar atau tidak berkepentingan), situs web yang masih dalam tahap perbaikan atau masih dibangun (Umumnya situs web akan memberikan halaman web yang berisi kata-kata seperti "*under construction*" atau "*coming soon*" dan situs web yang. Pada bulan januari 2018 terhitung bahwa terdapat 66 situs dari total 530 situs resmi pemerintah daerah yang masih belum atau tidak fungsional.

   Jika ditemukan bahwa sebuah situs web resmi pemerintahan daerah di Indonesia memiliki permasalahan mengenai situs web yang belum atau tidak fungsional maka situs tersebut akan dilewati pada pengambilan *main content* karena dianggap bahwa situs-situs tersebut belum atau tidak memiliki *main content* yang relevan atau sesuai dengan yang ingin didapatkan pada penelitian ini.
2. Situs web yang hanya berupa portal yang berisi link ke subdomain atau domain lain yang menyulitkan proses *crawling* dan proses pengambilan *main content* dimana pada bulan januari 2018 terhitung bahwa terdapat 23 situs dari total 530 situs resmi pemerintah daerah yang memiliki halaman *landing* dari *url* seperti ini.

   Situs web resmi pemerintah daerah di Indonesia memiliki halaman beranda atau *landing* yang hanya terdiri atas portal yang berisi *link* ke *subdomain* atau *domain* lain akan menyulitkan pengambilan *main content* dimana umumnya *link* yang mengarah ke *subdomain* adalah link menuju ke aplikasi milik pe-

merintah daerah misalnya LPSE dimana akan menyulitkan dan mengurangi akurasi pengambilan *main content*. Sehingga pada penelitian ini dibatasi bahwa halaman beranda atau halaman landing adalah memiliki setidaknya tujuh buah *link* yang mengarah pada *domain* yang sama dengan *url* resmi untuk setiap situs web resmi pemerintah daerah (pengecualian untuk www dimana www.surabaya.go.id dianggap sama dengan surabaya.go.id) dengan menghiraukan *link* yang mengarah ke *domain* lain termasuk link yang mengarah ke *subdomain* yang berada dibawah *url* resmi pemerintah daerah tersebut. Selain itu proses *crawling* pada situs web resmi pemerintah hanya dilakukan untuk *link-link* yang mengarah ke *domain* yang sama tanpa mengambil *link* yang mengarah ke *subdomain* atau *domain* lain.

3. Situs web yang dibangun dengan teknologi atau bahasa pemprograman yang menyulitkan untuk melakukan pengambilan data seperti penggunaan teknologi iframe. Pada bulan januari 2018 terhitung bahwa terdapat 39 situs dari total 530 situs resmi pemerintah daerah yang menggunakan teknologi iframe untuk menampilkan *main content* pada situs web mereka.

   Dikarenakan sifat dari iframe sendiri yang mengenkapsulasi konten dimana iframe sendiri dapat dikatakan sebagai halaman web terpisah sehingga apabila sebuah halaman web menggunakan iframe untuk menyajikan *main content* maka dapat dianggap bahwa pada halaman web tersebut terdapat halaman web lain yang berisi *main content*. Hal ini akan menyulitkan proses pengambilan *main content* pada penelitian ini.

Selain permasalahan teknikal, juga didapati bahwa situs web pemerintah daerah memiliki kekurangan dalam memberikan informasi mengenai pemerintah daerah masing-masing. Dengan mengambil kategori yang terdapat dalam Panduan Penyelenggaraan Situs Pemerintah pada bulan Juni 2017 ditemukan bahwa situs web resmi pemerintah daerah memiliki permasalahan mengenai:

1. Pada kategori selayang pandang yang memiliki berberapa item informasi seperti sejarah, motto, lambang, lokasi serta visi dan misi dengan masing-masing memiliki sebesar 58.39% untuk sejarah, 8.21% untuk motto, 50.73% untuk

lambang, 49.64% untuk lokasi dan 60.58% untuk visi dan misi dari total keseluruhan 530 situs web resmi pemerintah daerah di Indonesia yang menampilkan item informasi tersebut pada situs web mereka.

2. Pada kategori pemerintah darah yang menjelaskan mengenai struktur pemerintah daerah hanya memiliki 34.32% dari total keseluruhan 530 situs web resmi pemerintah daerah di Indonesia yang menampilkan informasi mengenai struktur pemerintah daerah.
3. Pada kategori geografi yang memiliki berberapa item informasi seperti topografi, demografi, cuaca dan iklim, sosial dan ekonomi serta budaya dengan masing-masing memiliki sebesar 41.79% untuk topografi, 16.97% untuk demografi, 21.17% untuk cuaca dan iklim, 23.91% untuk sosial dan ekonomi dan 38.87% untuk budaya dari total 530 keseluruhansitus web resmi pemerintah daerah di Indonesia yang menampilkan item informasi tersebut pada situs web mereka.
4. Pada kategori Peta Wilayah dan Sumberdaya yang menjelaskan mengenai batas administrasi wilayah dalam bentuk peta wilayah hanya memiliki 35.95% dari total keseluruhan 530 situs web resmi pemerintah daerah di Indonesia yang menampilkan informasi mengenai peta wilayah dan sumberdaya.
5. Pada kategori Peraturan/Kebijakan Wilayah yang menjelaskan mengenai Peraturan Daerah (Perda) yang dikeluarkan oleh pemerintah daerah tersebut hanya memiliki 34.31% dari total keseluruhan 530 situs web resmi pemerintah daerah di Indonesia yang menampilkan informasi mengenai struktur pemerintah daerah.

Dengan rata-rata kurang dari 60% untuk setiap informasi yang ditampilkan pada situs web resmi pemerintah daerah di Indonesia, terlihat bahwa pendisribusian informasi mengenai pemerintah daerah ke masyarakat masih kurang dan masih belum mencapai target sesuai dengan Instruksi Presiden Republik Indonesia no.3 Tahun 2003 tentang kebijakan dan strategi Nasional Pengembangan E-Government.

2.4 Pendefinisian *main content*

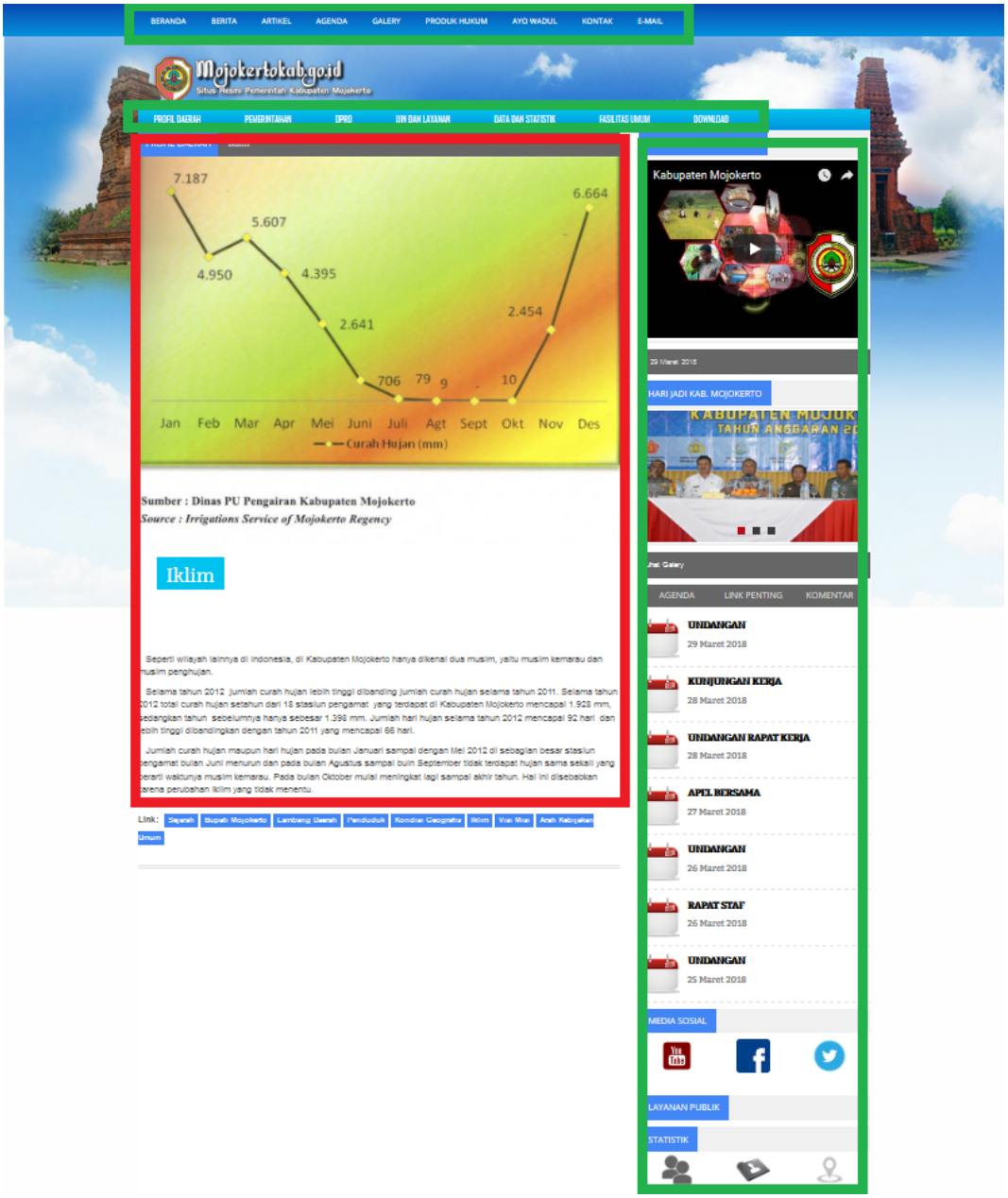
Organisasi W3 mendefinisikan *main content* sebagai konten yang unik pada dokumen tersebut dan tidak termasuk konten yang muncul berulang kalo pada sekumpulan dokumen seperti navigasi situs, informasi hak cipta, logo situs, banner dan formulir pencarian [w3 Organization, 2018]. Sedangkan Peter [Peters and Lecocq, 2013] mendefinisikan *main content* sebagai semua artikel teks termasuk judul, tanggal dan informasi penulis. Kohlschutter [Kohlschütter et al., 2010] menjelaskan bahwa blok *main content* seharusnya lebih "padat" dibandingkan blok yang bukan *main content*.

Pada penelitian ini *main content* didefinisikan sebagai sebuah bagian, segmen atau blok yang berisi konten yang berupa teks atau dalam bentuk multimedia yang berada pada sebuah halaman web yang bukan merupakan halaman web landing atau beranda seperti yang terlihat pada Gambar 2.12 dan bersifat unik pada satu halaman web didalam sebuah situs web dan tidak muncul berulang-ulang melebihi dari batas yang telah ditentukan pada setiap halaman web yang ada pada sebuah situs web. Contoh *main content* dapat dilihat pada gambar 2.13. Terlihat bahwa bagian yang ditandai dengan kotak merah pada gambar 2.13, jika dibandingkan dengan halaman web pada gambar 2.14 adalah bagian yang unik dan hanya ada pada halaman web pada gambar 2.13 dan hal yang sama juga dapat dilihat pada gambar 2.14 dimana bagian yang ditandai dengan kotak merah merupakan bagian yang unik dan hanya terdapat pada gambar 2.14. Dengan demikian ke dua bagian unik yang ditandai dengan kotak merah pada gambar 2.13 dan gambar 2.14 merupakan *main conten* pada kedua halaman web tersebut.

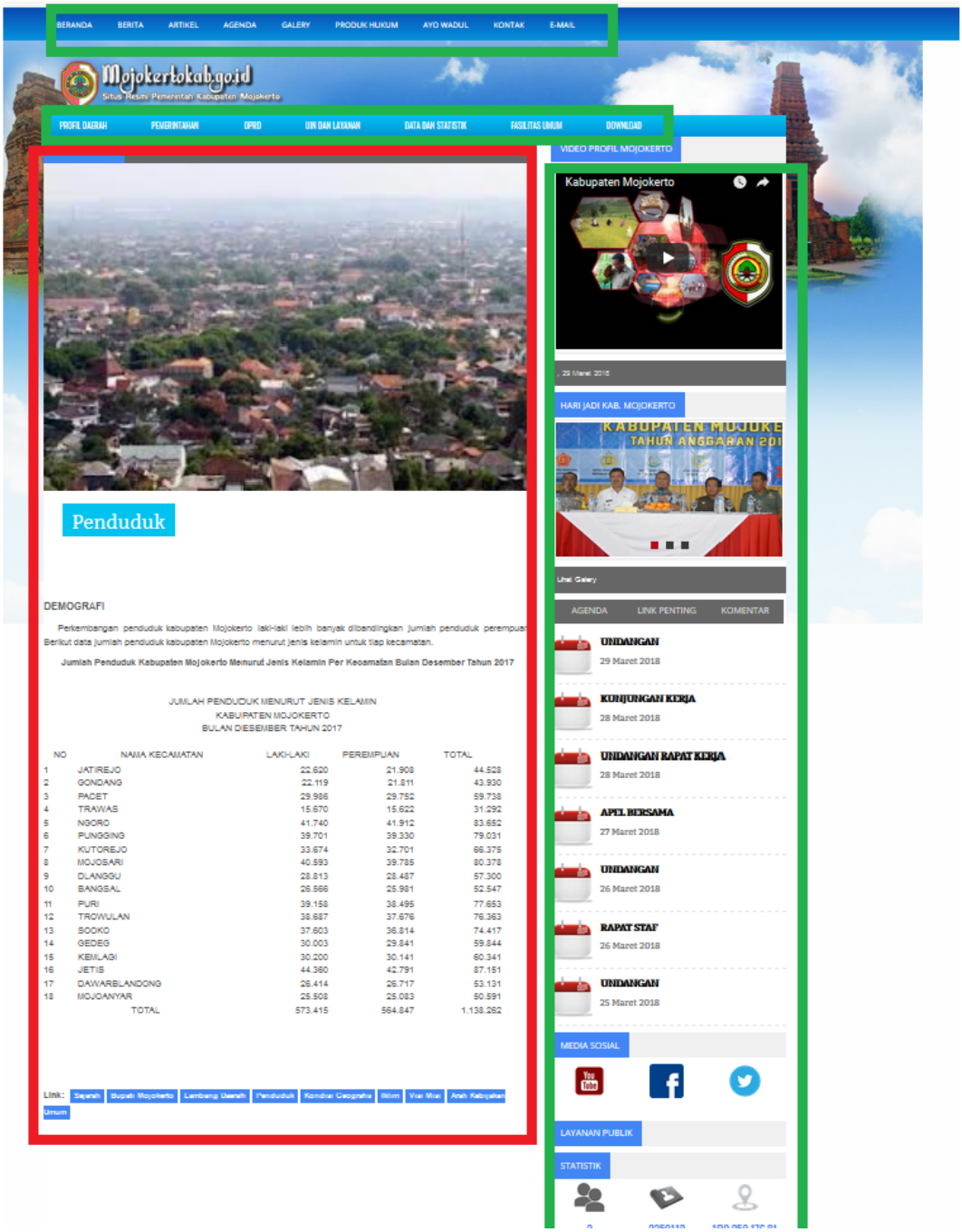
Organisasi w3.org mendefinisikan bahwa *main content* bersifat unik, sehingga untuk konten yang tidak bersifat unik atau muncul berulang kali dapat dikatakan sebagai *noisy content* [w3 Organization, 2018]. Barua mendefinisikan *noisy content* adalah konten yang muncul pada setiap artikel yang ada pada sebuah situs [Barua et al., 2014].

Pada penelitian ini *noisy content* didefinisikan sebagai sebuah bagian, segmen atau blok yang berisi konten yang berupa teks atau multimedia yang berada pada

Gambar 2.12: Halaman Beranda situs web resmi Pemerintah Daerah Kabupaten Mojokerto



Gambar 2.13: Contoh *main content* Ditandai Dengan Kotak Merah Pada Sebuah Halaman Web Pada situs web resmi Pemerintah Daerah Pemerintah Daerah Kabupaten Mojokerto



Gambar 2.14: Contoh *Noisy Content* Ditandai Dengan Kotak Hijau Pada Sebuah Halaman Web Pada situs Pemerintah Daerah Pemerintah Daerah Kabupaten Mojokerto

sebuah halaman web yang bukan merupakan halaman web landing atau beranda dan muncul berulang-ulang melebihi batas yang telah ditentukan pada setiap halaman web yang ada pada sebuah situs web. Contoh *noisy content* dapat dilihat pada gambar 2.13. Terlihat bahwa bagian yang ditandai dengan kotak hijau pada gambar 2.13, jika dibandingkan dengan halaman web pada gambar 2.14 adalah bagian yang muncul berulang pada gambar 2.13 dam gambar 2.14 sehingga bagian yang ditandai dengan kotak hijau merupakan *noisy content* pada kedua halaman web tersebut.

Halaman ini sengaja dikosongkan

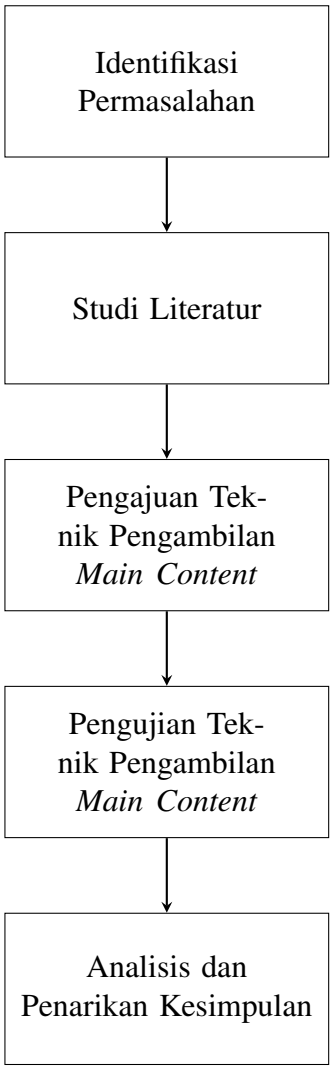
BAB 3

METODOLOGI PENELITIAN

Bab ini akan menjelaskan langkah-langkah yang diperlukan dalam proses penelitian sebagai kerangka acuan dalam proses pengerjaan tesis, sehingga rangkaian pengerjaan dapat dilakukan secara terarah, teratur, dan sistematis.

3.1 Tahapan Penelitian

Pada bagian berikut ini akan dijelaskan mengenai tahapan-tahapan yang akan dilakukan dalam penelitian ini seperti yang terlihat pada Gambar 3.1.



Gambar 3.1: Tahapan Penelitian

3.1.1 Identifikasi Permasalahan

Pengidentifikasian masalah bertujuan untuk menemukan *research question* yang ingin diselesaikan pada penelitian ini. Identifikasi permasalahan pada penelitian ini menggunakan pendekatan studi kasus telah dibahas pada bagian sebelumnya. Berdasarkan permasalahan yang telah teridentifikasi, tujuan yang ingin dicapai pada penelitian ini yaitu menghasilkan solusi dalam permasalahan pengambilan *main content* yang terdapat pada egovbench.

3.1.2 Studi Literatur

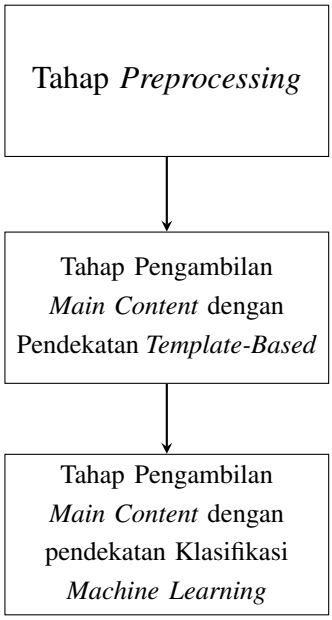
Studi literatur dalam penelitian ini bersumber dari buku, media, ataupun dari hasil penelitian orang lain. Pemahaman terhadap literatur ini bertujuan untuk menyusun dasar teori terkait yang digunakan dalam melakukan penelitian. Studi literatur yang digunakan pada penelitian ini adalah literatur yang berkaitan dengan pengambilan *main content*.

3.1.3 Pengajuan dan Formulasi Pengambilan *Main Content*

Pada Tahap ini akan diajukan mengenai definisi dan pendekatan yang diajukan dalam melakukan pengambilan *main content* pada situs web resmi pemerintah daerah di Indonesia. Berdasarkan hasil studi literatur, pada penelitian akan diajukan mengenai pengambilan *main content* dengan berberapa langkah seperti yang terlihat pada gambar 3.2.

Garis besar langkah-langkah dalam Pengambilan *Main Content* pada penelitian ini dijabarkan sebagai berikut:

1. Dengan berdasarkan *link* url dari situs resmi pemerintah dilakukan tahapan *preprocessing* dilakukan . Tahapan *preprocessing* ini dilakukan untuk mendapatkan *input* atau masukan yang valid dan dapat diproses pada tahap pengambilan *main content*. Hasil dari tahap preprocessing ini adalah *preprocessed content* dimana konten sudah dapat diproses pada tahap pengambilan *main content*.
2. Preprocessed Content yang dihasilkan pada tahap preprocessing kemudian di-



Gambar 3.2: Tahapan Pengambilan *Main content*

lakukan pengambilan main content dengan pendekatan template-based untuk membentuk kandidat *main content*.

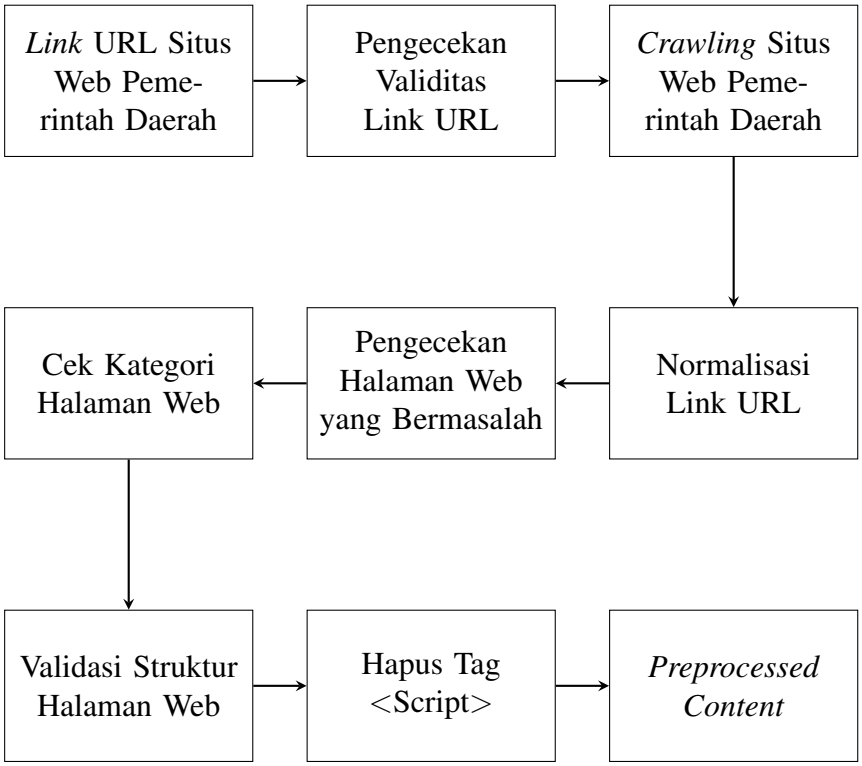
3. Pengambilan *main content* dengan pendekatan klasifikasi *machine learning* dilakukan pada kandidat *main content* untuk menentukan apakah blok atau *tag* tersebut merupakan *main content* atau bukan *main content*.

3.1.3.1 Tahap *Preprocessing*

Tujuan dari tahap *preprocessing* ini adalah untuk memastikan bahwa *link url* yang digunakan adalah *link url* yang valid dan halaman web dapat diproses pada tahap pengambilan *main content* untuk mendapatkan *main content* dari halaman web pemerintah daerah di Indonesia. Tahapan *preprocessing* yang dilakukan pada penelitian ini dapat dilihat pada gambar 3.3.

Tahapan *preprocessing* pada penelitian ini secara garis besar terbagi menjadi berberapa langkah yaitu:

1. Pengambilan *link url* untuk situs resmi pemerintah daerah di Indonesia dimana *link url* yang digunakan adalah link url yang sesuai pada situs Kementrian Komunikasi dan Informasi Indonesia dan sesuai dengan domain go.id.
2. Pengecekan validitas dari *Link Url* yang ada pada situs web pemerintah dae-



Gambar 3.3: Tahapan *Preprocessing*

rah dan untuk memastikan bahwa *link url* resmi dari pemerintah daerah merupakan *link* yang valid dan dapat dilakukan *crawling* pada *link url* tersebut untuk mencari halaman web yang ada pada situs pemerintah daerah tersebut.

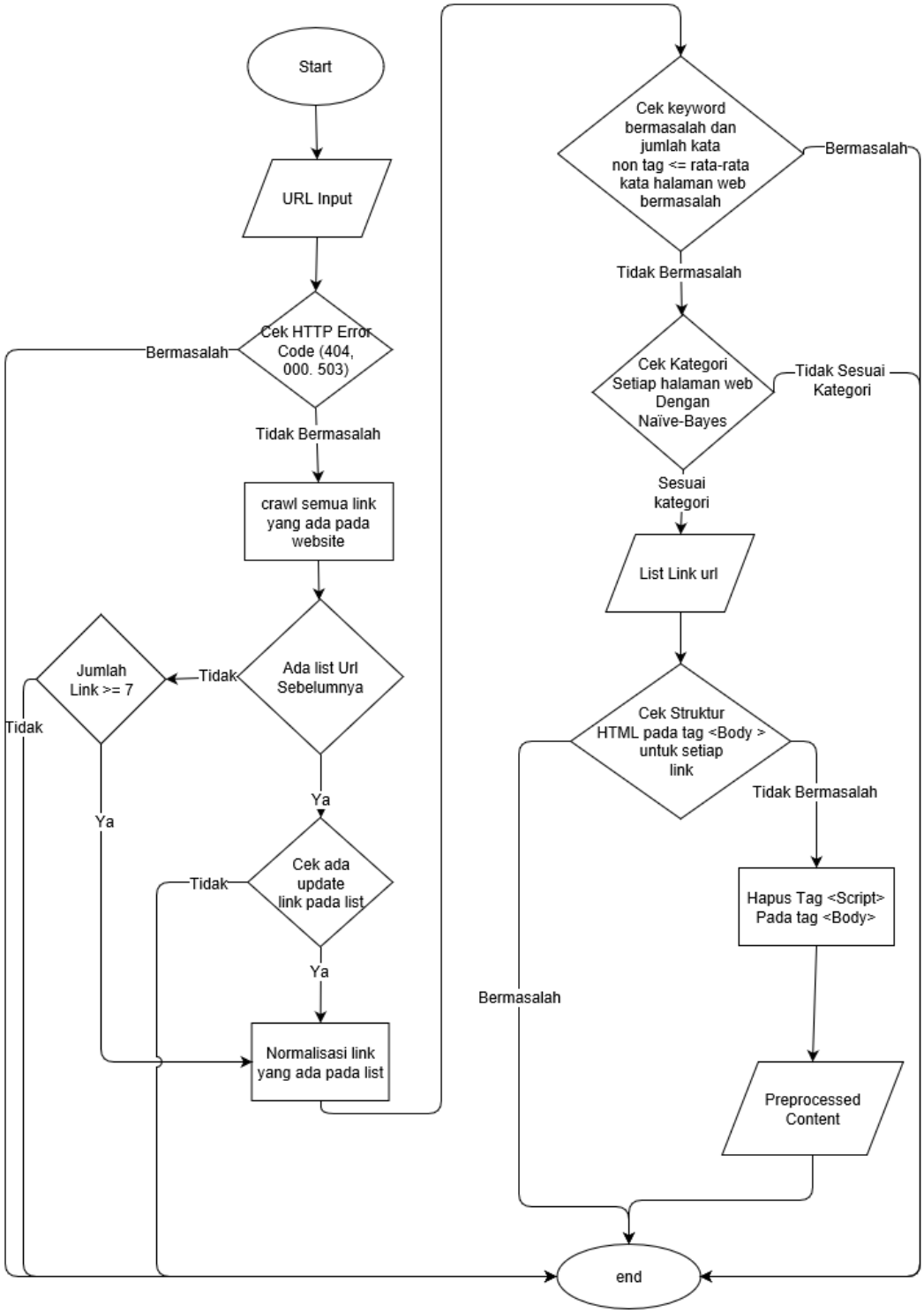
3. Proses *crawling* dilakukan untuk menemukan semua *link* yang ada pada situs resmi pemerintah daerah tersebut.
4. Normalisasi *link url* dilakukan untuk menghindari link url yang tidak normal seperti link yang berbentuk seperti"\home \" daripada link normal yang berbentuk seperti "http://www.surabaya.go.id/home". selain itu normalisasi dilakukan untuk melihat *link* url yang duplikat atau *link* url tersebut sudah pernah dikunjungi sebelumnya.
5. Pengecekan halaman web yang bermasalah dilakukan untuk menyaring halaman web yang dapat diakses akan tetapi tidak memiliki konten yang dapat diambil seperti halaman web yang masih dalam perbaikan dan berisi tulisan "*under construction*" atau "*coming soon*".
6. Pengecekan kategori halaman web dilakukan untuk mengkategorisasikan ha-

laman web sesuai dengan kategori Panduan Peneyelenggaraan Situs Pemerintah Daerah

7. Pengecekan atau validasi struktur HTML tag pada halaman web untuk mengetahui apakah struktur HTML atau HTML *markup* sebuah halaman web adalah struktur yang valid dan dapat diproses pada tahap pengambilan *main content*.
8. Penghapusan tag <script> pada halaman web untuk meningkatkan akurasi karena isi dari tag <script> umumnya adalah hal yang tidak relevan dengan *main content*
9. Hasil dari tahap *preprocessing* dimasukkan kedalam *preprocessed content* untuk diproses pada tahap pengambilan *main content*.

Berdasarkan tahapan *preprocessing* maka dibentuk alur dari *preprocessing* yang dapat dilihat pada gambar 3.4. Alur *preprocessing* tersebut terbagi menjadi berberapa langkah yang memperlihatkan bagaimana proses *preprocessing* mulai dari memproses *link url* resmi pemerintah daerah hingga mendapatkan *preprocessed content*. Langkah-langkah tersebut yaitu:

1. Dengan berdasarkan *link url* resmi dari pemerintah daerah (contoh : http://ponorogo.go.id) maka akan dilakukan pengecekan apakah *link url* tersebut adalah *link* yang valid dan dapat diakses dengan melakukan pengecekan terhadap respon *http status code*. jika respon *http status code* adalah *status* 404, *status* 000 atau *status* 503, menandakan bahwa *link* tersebut tidak valid atau tidak dapat diakses dan proses berhenti.
2. Apabila *link url* resmi dari pemerintah daerah adalah *link url* valid dan dapat diakses, maka selanjutnya akan dilakukan crawling untuk mengambil semua *link* yang ada pada halaman web *landing* atau beranda dari situs web resmi pemerintah daerah tersebut. link yang akan diambil disini adalah link yang mengarah kepada *domain* yang sama dengan halaman landing atau beranda. Halaman beranda pada penelitian ini didefinisikan sebagai *halaman landing* dari *link url* resmi pemerintah daerah yang memiliki setidaknya tujuh buah *link* yang mengarah pada *domain* yang sama dengan *url* resmi untuk se-



Gambar 3.4: Alur *Preprocessing*

tiap situs web resmi pemerintah daerah (pengecualian untuk www dimana www.surabaya.go.id dianggap sama dengan surabaya.go.id) dengan menghiraukan *link* yang mengarah ke *domain* lain termasuk *link* yang mengarah ke *subdomain* yang berada dibawah *url* resmi pemerintah daerah tersebut. Selain itu proses *crawling* pada situs web resmi pemerintah hanya dilakukan untuk *link-link* yang mengarah ke *domain* yang sama tanpa mengambil *link* yang mengarah ke *subdomain* atau *domain* lain.

Selain aturan mengenai pembatasan *crawling* hanya ke *domain* utama, Halaman beranda juga harus diperlukan untuk memiliki setidaknya 7 buah link atau lebih ke *domain* yang sama. batasan minimal tujuh buah link ini didapatkan dengan melihat bahwa menurut Panduan Penyelenggaraan Situs Pemerintah Daerah [dan Informasi, 2009] terdapat 7 kategori informasi yang harus ada yaitu:

(a) selayang pandang
(b) struktur pemerintah daerah
(c) geografi
(d) peta wilayah dan sumberdaya
(e) peraturan dan kebijakan wilayah
(f) berita
(g) pesan dan saran

3. Ketika proses *crawling link* telah selesai dilakukan, maka selanjutnya akan dilihat apakah sudah ada *list url* yang telah ada sebelumnya. jika tidak ada list url sebelumnya atau proses *crawling* pertama kali maka dilihat apakah hasil *crawling* dari halaman beranda memiliki setidaknya 7 buah link sesuai dengan definisi dari halaman beranda. Sedangkan jika telah ada *list url* sebelumnya maka dilihat apakah terdapat perubahan atau perbedaan antara *list url* yang baru dengan telah ada.
4. Dengan terbentuknya *list url* maka selanjutnya dilakukan proses normalisasi untuk setiap *url*. Normalisasi *link url* dilakukan untuk menghindari link url yang tidak normal seperti link yang berbentuk seperti"\home \" daripada

link normal yang berbentuk seperti ”http://www.surabaya.go.id/home”. selain itu normalisasi dilakukan untuk melihat *link* url yang duplikat atau *link* url tersebut sudah pernah dikunjungi sebelumnya.

5. Setelah proses normalisasi kemudian akan dilihat pada setiap halaman web apakah mengandung *keyword* yang bermasalah yaitu kata-kata seperti suspended, perbaikan, , hack, *construction*, *it works*, maaf, beranda,*temporarily*, masuk, *hostname*, dan *forbidden*. Dimana kata-kata tersebut umumnya menandakan bahwa halaman web tersebut belum atau tidak fungsional. Contohnya tahap pembuatan yang akan memberikan kata-kata seperti *under construction* atau *coming soon*. Selain *keyword* juga digunakan jumlah kata-kata yang bukan merupakan *tag html* kurang dari atau sama dengan rata-rata jumlah kata pada halaman web yang bermasalah. Batasan ini dilakukan karena umumnya halaman web yang mengandung kata-kata tersebut dan memiliki jumlah kata kurang dari 20 merupakan halaman web bermasalah atau sedang dalam proses pembuatan sehingga tidak memiliki *main content*.

   Selain itu juga dilakukan penyaringan terhadap kata-kata di dalam *tag* HTML seperti iframe, http-equiv=”refresh” atau location. Penyaringan terhadap kata iframe dilakukan karena konten terenkapsulasi oleh iframe dan diambil pada tempat atau *link* lain sehingga menyulitkan pengambilan *link url*. Sedangkan penyaringan terhadap kata http-equiv=”refresh” atau location dilakukan karena kata-kata tersebut merupakan contoh penggunaan cara yang tidak biasa dalam melakukan *url redirection* yang menyulitkan pengambilan *link url*.
6. Untuk setiap *link* yang melewati tahap sebelumnya maka kemudian akan dilakukan proses *machine learning* dengan naïve-bayes yang dibangun oleh Wisnu [Sugiyanto, 2017] sesuai dengan kategori informasi pada Panduan Penyelenggaraan Situs Pemerintah Daerah untuk menentukan apakah *link url* tersebut termasuk kedalaman tujuah kategori menurut Panduan Penyelenggaraan Situs Pemerintah Daerah.
7. Untuk setiap *link* yang ada kemudian akan diambil isi dari tag <body> dan dilakukan pengecekan apakah terdapat *tag* HTML yang tidak sesuai atau hi-

lang. Pengecekan dan validasi terhadap struktur atau HTML *markup* pada setiap *link* atau halaman web dilakukan agar sebuah halaman web dapat diproses pada tahap pengambilan *main content*. Pada tabel 3.1 adalah berberapa kondisi untuk melakukan validasi terhadap struktur HTML pada sebuah halaman web [Raggett, 1998, Park et al., 2013].

Jika ditemui sebuah halaman web memiliki salah satu atau lebih dari kondisi yang terdapat pada tabel 3.1, maka halaman web tersebut akan dilewati atau tidak dilakukan pengambilan *main content* karena pada penelitian ini pada tahap pembentukan template pada pengambilan main content akan sangat terpengaruh dengan struktur DOM dari sebuah halaman web jika struktur sebuah halaman web memiliki kekurangan atau ketidaklengkapan seperti kondisi-kondisi tersebut maka akan dapat mengakibatkan kesalahan dalam pembentukan DOM.

Selain itu Samimi mengatakan bahwa dalam memperbaiki HTML *generation error* akan ditemui lebih dari satu macam pilihan [Samimi et al., 2012] dalam memperbaiki HTML *generation error* tersebut dan berbagai pilihan dalam memperbaiki ini adalah bersifat menerka dari ”hal yang seharusnya dari pandangan pihak developer”. Sifat menerka ini akan dapat mengurangi akurasi dalam pengambilan *main content* sehingga diputuskan bahwa halaman web dengan permasalahan struktur maka akan dilewati.

Selain itu juga dibatasi bahwa untuk tag <body> yang diambil adalah tag <body> yang setidaknya memiliki minimum satu buah *child node* atau *tag* HTML yang berada tepat satu level di bawah tag <body>. Hal ini dilakukan karena pada tahap pembuatan *template* pada tahap pengambilan *main content* setidaknya membutuhkan satu buah *node* untuk melakukan komparasi pada berberapa halaman web.

8. Jika halaman web dinyatakan tidak memiliki permasalahan pada *tag* html maka selanjutnya akan dilakukan penghapusan tag <script> yang ada pada halaman web. Penghapusan tag <script> ini dimaksudkan untuk meningkatkan akurasi karena isi dari tag <script> umumnya adalah hal yang tidak relevan

dengan *main content*

9. Setelah tahap-tahap sebelumnya dilakukan, maka hasil dari proses *preprocessing* untuk setiap halaman web akan dimasukkan kedalam *list preprocessed content* yang akan digunakan pada tahap selanjutnya dalam proses pengambilan *main content*

Tabel 3.1: Rule Untuk validasi struktur HTML

Error Type	Example
Mismatched end tags	subheading
Misnested tags	boldbold italic bold?
Missing end tag / Unclosed Pairs	<h1>heading
Mixed-up tags	<p>new paragraph bold text <p>some more bold text
Missing "/" in end tags	<h1><hr>heading<h1>
List markup with missing tags	<body> 1st list item 2nd list item
Tags without terminating >	<h1><hr>heading</h1
Typographical Error	<dov>heading</dov>
Structural Fatal Error	<html></html>

3.1.3.2 Tahap Pengambilan *Main Content* dengan Pendekatan *Template-Based*

Pada tahap ini untuk pengambilan *main content* pada halaman web dilakukan pembentukan *template* untuk mengambil blok atau segment yang kemungkinan besar merupakan *main content*. Pada penelitian ini untuk membentuk *template* dari halaman web yang akan dilakukan pengambilan *main content* diajukan sebuah algoritma *Multiple Restricted Top-Down Mapping*. Algoritma *Multiple Restricted Top-Down Mapping* ini diajukan dengan melihat bahwa algoritma *restricted top-down mapping* yang diajukan oleh viera melakukan komparasi secara berpasang-pasangan dari kumpulan halaman web yang diambil (misal halaman web a dan b,a dan c, b dan c) untuk membangun *template* dari situs web [Vieira et al., 2006]. Selain itu algoritma *restricted top-down mapping* mengambil secara acak halaman web yang ada pada situs web untuk membuat *template*. Pada Penelitian diajukan Algoritma *Multiple Restricted Top-Down Mapping* yang berusaha untuk dapat me-

lakukan komparasi berberapa halaman web secara sekaligus. Berberapa istilah yang digunakan pada penelitian ini adalah:

(a) Node : element yang ada pada dokumen HTML.

(b) *Top Node*: *Node* yang akan dilakukan pencarian untuk *child node* seperti yang terlihat pada Gambar 3.5.

(c) *Child Node*: *Element Node* yang merupakan *branches* dari *top node* pada DOM *tree* seperti yang terlihat pada Gambar 3.5.

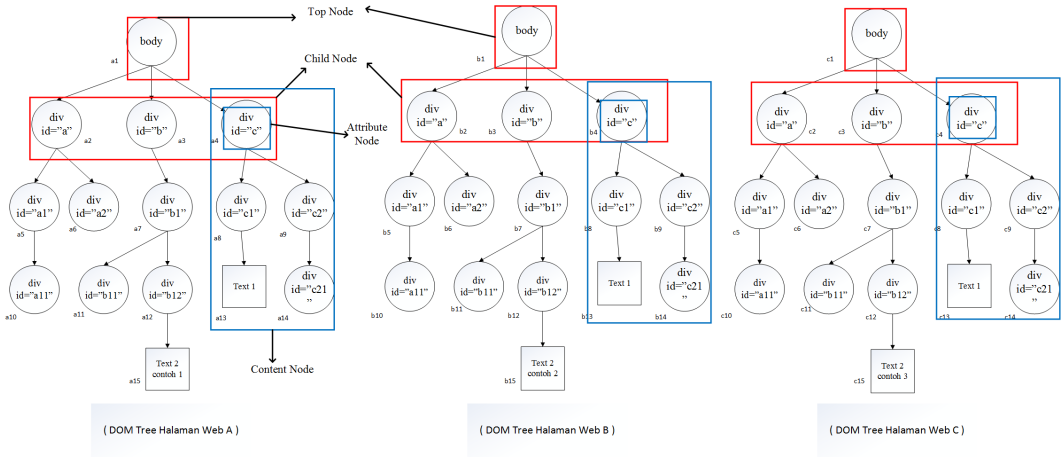
(d) *Attribute Node*: attribut dari sebuah *node* seperti yang terlihat pada Gambar 3.5.

(e) *Content Node*: konten dari sebuah *node* seperti yang terlihat pada Gambar 3.5.

(f) *Frequent Node*: *node* yang memiliki kemunculan sama dengan atau melebihi batas atau *threshold* yang ditentukan pada kumpulan *node*. Contohnya jika melihat pada Gambar 3.5 dengan *threshold* yang ditentukan adalah 2 maka untuk operasi *Frequent Node* pada kumpulan node a2,a3,a4danb2,b3,b4 dengan *attribute node* div id=”a”, div id=”b”, div id=”c”, div id=”a”, div id=”b”, div id=”c” adalah div id=”a”, div id=”b”, div id=”c” karena masing-masing mempunyai nilai kemunculan berupa 2 yang merupakan sama dengan *threshold* yang telah ditentukan.

Berikut ini adalah langkah-langkah yang dilakukan pada Algoritma *Multiple Restricted Top-Down Mapping*:

1. Dilakukan penentuan *top node*, jika ini adalah iterasi pertama maka *top node* adalah HTML tag <body>.
2. Dilakukan pencarian *child node* dari *top node* tersebut.
3. *Content node* dan *attribute node* ditentukan untuk setiap *child node*, jika tidak ditemukan *child node* pada *top node* maka *content node* dan *attribute node* dianggap kosong.
4. Pencarian *frequent node* dilakukan untuk *content node* dan *attribute node* yang telah dibuat.
5. Untuk setiap *child node* yang memiliki:



Gambar 3.5: Struktur DOM tree Halaman web A dan B

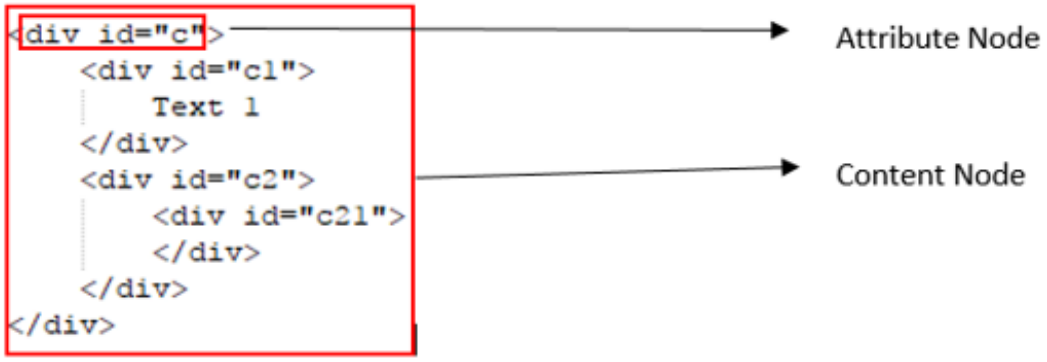
(a) *content node* dan *attribute node* kurang dari *threshold* atau batas yang ditentukan maka tandai sebagai kandidat *main content*.

(b) *content node* kurang dari *threshold* atau batas yang ditentukan dan *attribute node* lebih besar dari *threshold* atau batas yang ditentukan, maka set *node* tersebut menjadi *top node* kemudian ulangi langkah 2.

6. *Template* dibentuk dengan menggabungkan semua *node* yang tidak ditandai sebagai kandidat *main content*.

Pada Gambar 3.5 terlihat 3 struktur DOM tree dari halaman web A, halaman web B dan halaman web C, dimana halaman web A, halaman web B, halaman web C berada dibawah satu website yang sama. Bentuk bundar mengindikasikan bahwa hal tersebut adalah *node* atau *tag* sedangkan bentuk persegi mengindikasikan bahwa hal tersebut adalah teks atau konten dari *tag* diatasnya.

Selanjutnya *top node* pertama akan dipilih dimana pada contoh ini *top node* pertama adalah *node* paling atas atau *node body*. Dari *node body* tersebut kemudian dilanjutkan dengan pencarian *child node* dari *top node* tersebut seperti yang terlihat seperti Gambar 3.5. Setelah *child node* ditentukan, kemudian untuk setiap *child node* yang tersebut dilakukan 2 buah operasi *Frequent node* terpisah dimana setiap *content node* dan *attribute node* diambil dari setiap *child node*. Operasi *Frequent node* yang pertama yaitu dengan menggunakan isi konten dari setiap *child node* yang dinamakan *content node*. Sedangkan Operasi *Frequent node* yang kedua



Gambar 3.6: HTML *tag* untuk *tag* div id=”c”

dengan menggunakan *attribute* dari *child node* tersebut yang dinamakan *attribute node*. Ilustrasi dari *content node* dan *attribute node* dapat dilihat pada Gambar 3.5. Untuk lebih jelasnya dari gambar Gambar 3.5, untuk *tag* div id=”c” jika dirubah menjadi bentuk HTML *tag* maka akan tampak seperti Gambar 3.6.

Setelah dilakukan operasi *frequent node* terhadap *attribute node* dan *content node* maka dilihat *node* mana yang memiliki ketidakmiripan dengan halaman web lain atau dapat dikatakan tidak mencapai batas atau *threshold* frekuensi kemiripan yang telah ditentukan.

Untuk setiap *child node*, jika *content node* dan *attribute node* memiliki frekuensi yang sama dengan atau diatas batas atau *threshold* yang ditentukan maka *child node* tersebut selalu ada pada setiap halaman web pada sebuah situs tersebut sehingga *child node* tersebut memiliki kemungkinan yang besar adalah *template*. Apabila *content node* dan *attribute node* memiliki frekuensi yang berada dibawah batas atau *threshold* yang ditentukan maka node tersebut adalah node yang sering berubah untuk setiap halaman web pada sebuah situs, sehingga *node* tersebut kemungkinan besar adalah *main content* dari halaman web tersebut yang selanjutnya akan kita tandai sebagai *main content*.

Akan tetapi, jika *child node* tersebut memiliki frekuensi *attribute node* yang sama dengan atau diatas batas atau *threshold* sedangkan *content node* memiliki frekuensi dibawah batas atau *threshold* maka dapat diambil kesimpulan bahwa terdapat *node* dibawahnya yang sering berubah pada setiap halaman web pada sebuah situs.

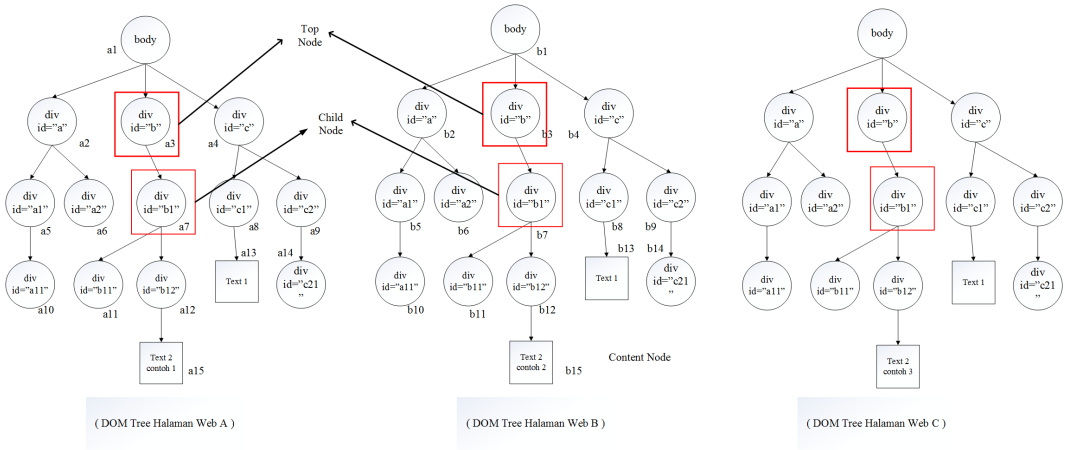
Sehingga *node* tersebut kemudian akan kita set menjadi *top node* menggantikan *top node* yang lama. Selanjutnya dilakukan pengecekan apakah *node* tersebut memiliki *child* atau tidak. Jika *node* tersebut tidak memiliki *child* untuk sebuah halaman web maka untuk *content node* dan *attribute node* dianggap kosong. Namun apabila node tersebut memiliki *child node* maka untuk masing-masing *child node*, *content node* dan *attribute node* akan di set sesuai dengan *child node* tersebut. Setelah *content node* dan *attribute node* telah ditentukan maka dilakukan proses yang sama menggunakan *frequent node*.

Hal ini terus dilakukan berulang kali sampai ditemukan kondisi *content node* dan *attribute node* memiliki frekuensi yang berada dibawah batas atau *threshold* yang ditentukan dan tidak *node* lain yang memiliki frekuensi *attribute node* yang sama dengan atau diatas batas atau *threshold* sedangkan *content node* memiliki frekuensi dibawah batas atau *threshold*. Selanjutnya *node* tersebut ditandai sebagai kandidat *main content*.

Setelah proses ini berhenti, untuk setiap *node* yang tidak ditandai sebagai kandidat *main content* akan ditandai *node* menjadi sebuah template pada situs web tersebut. Sedangkan *node* yang ditandai sebagai kandidant *main content* kemudian akan diproses pada tahap selanjutnya yaitu tahap klasifikasi *main content* untuk menentukan *node* tersebut merupakan *main content* atau bukan *main content*.

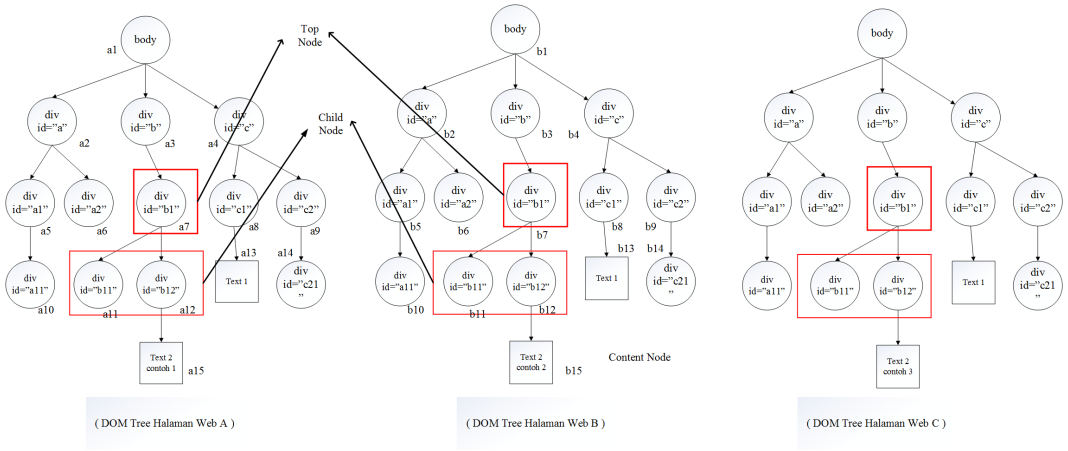
Pada gambar 3.5 dapat dilihat untuk *node* div id=”b” memiliki ketidakmiripan *content node* antara halaman web A, halaman web B dan halaman web C, dimana untuk *node* paling bawah halaman web A berisi “Text 2 contoh 1” , halaman web B berisi “Text 2 contoh 2” dan halaman web B berisi “Text 2 contoh 3”. Akan tetapi, untuk node tersebut memiliki attribute node yang mirip antara halaman web A, halaman web B dan halaman web C sehingga dapat disimpulkan terdapat sebuah *node* dibawahnya yang berubah untuk setiap halaman web.

Selanjutnya *node* div id=”b” akan diset menjadi *top node* dan dilakukan pencarian terhadap *child* dari *node* tersebut. *Child* yang ditemukan adalah *node* div id=”b1” untuk halaman web A, *node* div id=”b1” untuk halaman web B dan , *node* div id=”b1” untuk halaman web C seperti yang terlihat pada Gambar 3.7. Setelah



Gambar 3.7: Pencarian *child node* untuk *tag* div id=”b”

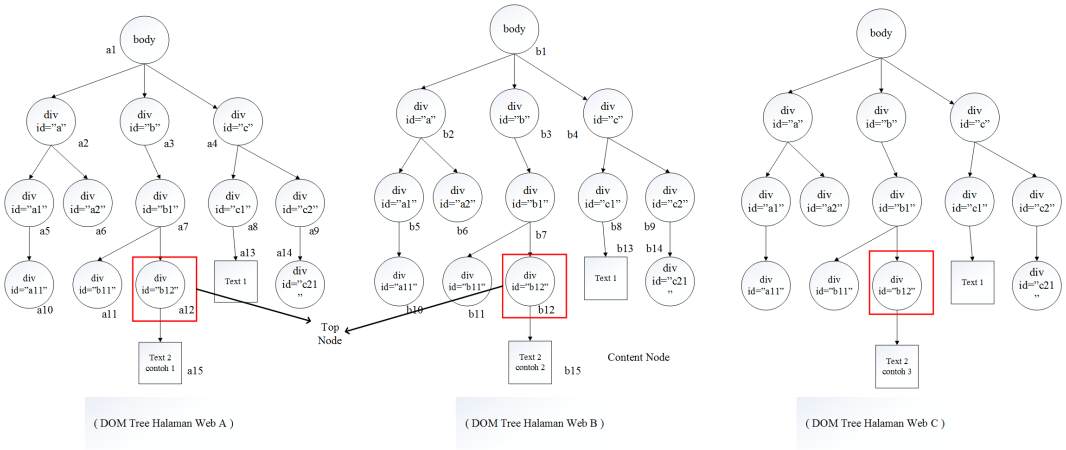
dilakukan *frequent node* ditemukan bahwa *node* div id=”b1” masih memiliki ketidakmiripan *content node*, akan tetapi memiliki kemiripan *attribute node*. Sehingga div id=”b1” di set menjadi *top node*.



Gambar 3.8: Pencarian child node untuk *tag* div id=”b1”

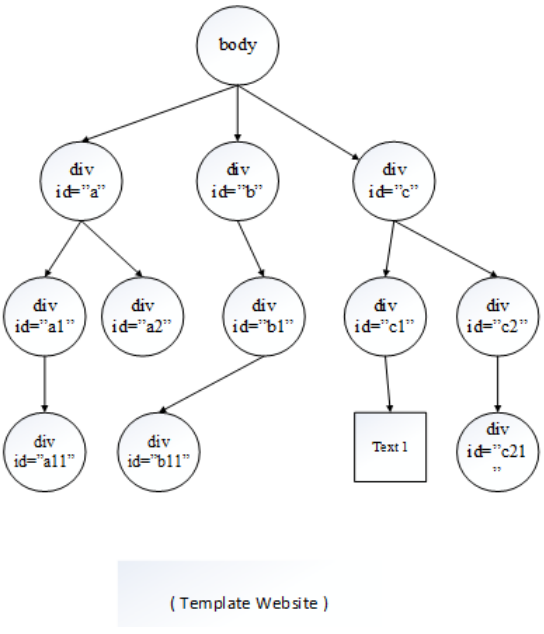
Sama halnya dengan proses sebelumnya seperti yang terlihat pada gambar Gambar 3.8, dilakukan proses yang sama mulai dari menentukan *child* dan melakukan operasi *frequent node*. Setelah dilakukan proses *frequent node* ditemukan bahwa untuk *node* div id=”b12” memiliki *content node* yang berbeda namun masih memiliki *attribute node* yang sama seperti yang terlihat pada gambar 3.9.

Kemudian *node* div id=”b12” di set menjadi *top node*, dan dilakukan pencarian terhadap *child* dari *node* div id=”b12”. Akan tetapi karena *node* div id=”b12” tidak memiliki *child* maka untuk masing halaman web A dan halaman web B, maka



Gambar 3.9: Pencarian child node untuk *tag* div id=”b12”

content node dan *attribute node* berisi akan berisi himpunan kosong yang kemudian jika dilakukan *frequent node* ditemukan *attribute node* dan *content node* akan memiliki hasil yang berupa tidak ada frekuensi *attribute node* yang sama dengan atau diatas batas atau *threshold*, maka *node* div id=”b12” ditandai sebagai kandidat *main content* dan proses ini berhenti.



Gambar 3.10: Hasil template situs yang terbentuk

Selanjutnya untuk menentukan *node* yang ditandai sebagai kandidat main content merupakan *main content* atau bukan *main content*, maka *node* tersebut kemudian diproses pada tahap klasifikasi Machine Learning untuk menentukan *main con-*

tent.

3.1.3.3 Tahap Pengambilan *Main Content* dengan pendekatan Klasifikasi Machine Learning

Pada tahap ini setiap *node* yang ditandai sebagai kandidat main content pada tahap pengambilan *main content* dengan pendekatan *template-based* dianggap sebagai sebuah blok atau segmen unik pada sebuah halaman web. Untuk menentukan sebuah blok atau segmen kandidat *main content* tersebut sebagai *main content* dilakukan pendekatan klasifikasi *machine learning*. *Dataset* untuk melakukan klasifikasi *machine learning* dibangun dari *dataset* yang dibangun dari tahap pengambilan *main content* menggunakan pendekatan *template-based*. Dari keseluruhan dataset yang terbentuk tersebut, 70% dari dataset akan digunakan untuk tahap *training data* dan 30% dari *dataset* akan digunakan sebagai tahap *testing classifier*.

Untuk setiap blok atau tag pada dataset tersebut akan dilakukan perhitungan *feature set* untuk menentukan apakah blok atau *tag* tersebut adalah *main content* atau tidak. *Feature Set* yang digunakan adalah feature set yang telah dimodifikasi dari penelitian yang dilakukan oleh Yao [Yao and Zuo, 2013] yaitu:

1. jumlah kata dalam blok.
2. Jumlah kalimat di blok.
3. Kepadatan teks (text density) di blok.
4. Jumlah link di blok.

Modifikasi *feature set* dilakukan dikarenakan pada penelitian Yao, perhitungan untuk setiap *feature set* juga mempertimbangkan kode atau *tag* HTML. Sebagai contoh jumlah kata pada penelitian Yao juga menghitung setiap kode atau *tag* HTML sebagai sebuah kata individual, sedangkan pada penelitian ini jumlah kata hanya menghitung kata yang muncul kepada user dan menghiraukan kode HTML atau *script*.

Selain itu, perubahan terbesar dilakukan pada *feature set* kepadatan teks di blok dimana Yao mencantumkan bahwa kepada teks dinilai dari jumlah kata dibandingkan dengan jumlah baris dimana jumlah kata juga mewakili jumlah *tag* atau kode HTML yang ada pada setiap line. Pada penelitian ini, kepadatan teks didefini-

sikan sebagai jumlah kata yang bukan kode HTML atau kode lainnya dibandingkan dengan jumlah *tag* yang ada pada blok tersebut. Perbandingan dengan *tag* dilakukan dengan melihat bahwa *tag* dimana berdasarkan definisi dari W3C [W3C, 2018] merupakan *tag* yang digunakan untuk mendefinisikan sebuah bagian pada sebuah dokumen HTML. Bagian ini umumnya merupakan menjadi pembatas untuk *tag-tag* HTML yang memiliki tujuan dan fungsi yang sama.

Selain *feature set* diatas, juga ditambahkan prediksi mengenai kategori *main content* pada halaman web pemerintah daerah yang dibangun pada penelitian yang dilakukan oleh Wisnu [Sugiyanto, 2017]. model prediksi yang dibangun oleh wisnu tersebut dibuat dengan berdasarkan kriteria-kriteria informasi pada situs pemerintah daerah yang ada pada Panduan Penyelenggaraan Situs Pemerintah Daerah yang diterbitkan oleh Departemen Komunikasi dan Informasi.

Setiap blok atau tag kemudian akan dilabeli menjadi *main content* atau *bukan main content* oleh *annotator* manusia. *Classifier* yang digunakan untuk membangun model dan melakukan klasifikasi adalah *classifier* naïve-bayes.

3.1.4 Pengujian Hasil Pengambilan *main content*

Pengujian hasil pengambilan *main content* pada penelitian ini dilakukan untuk mengetahui bagaimana kinerja yang dimiliki oleh pendekatan yang dilakukan penelitian ini terutama dalam mengenai akurasi dalam menentukan *main content*. Pengujian akan dilakukan berberapa kali dengan setiap pengujian memiliki konfigurasi yang berbeda satu sama lain, yaitu:

1. Pengujian dengan hanya menggunakan pembagian blok hasil pendekatan *template-based* untuk melakukan pengambilan *main content*.
2. Pengujian dengan menggunakan blok hasil pendekatan *template-based* dan klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi untuk melakukan pengambilan *main content*.
3. Pengujian dengan menggunakan blok hasil pendekatan *template-based*, klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi dan model prediksi kategori *main content* pada halaman web pemerintah da-

erah untuk menentukan *main content* yang dibangun oleh Wisnu [Sugiyanto, 2017].

Pengujian dengan model prediksi kategori *main content* pada halaman web pemerintah daerah untuk menentukan *main content* yang dibangun oleh Wisnu dilakukan karena pada saat ini egovbench menggunakan model tersebut untuk memprediksi kategori dari halaman web pada situs web resmi pemerintah daerah di Indonesia untuk melakukan penilaian. Model tersebut dibangun berdasarkan kriteria-kriteria informasi pada situs pemerintah daerah yang ada pada Panduan Penyelenggaraan Situs Pemerintah Daerah yang diterbitkan oleh Departemen Komunikasi dan Informasi. Sehingga, dengan dilakukan pengujian tersebut dapat diketahui bagaimana hasil prediksi pada kategori halaman web ketika menggunakan pendekatan yang dilakukan pada penelitian ini.

Penilaian akurasi pada pengujian akan dilakukan dengan 2 tahap yaitu pengukuran menggunakan *confusion matrix* yang akan digunakan pengukuran *accuracy*, *precision*, *recall* dan *f1-score* dan membandingkan hasil klasifikasi dengan evaluator manusia.

3.1.5 Analisis dan Penarikan Kesimpulan

Tahapan terakhir dalam penelitian ini yaitu menganalisis dan membahas secara menyeluruh temuan dan kinerja pada formulasi yang diajukan untuk pengambilan *main content* pada penelitian ini dan terkait dengan egovbench.

Halaman ini sengaja dikosongkan

BAB 4

HASIL DAN PEMBAHASAN

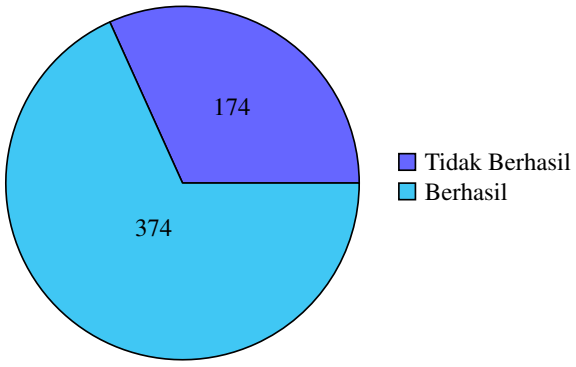
Pada bab ini akan dijelaskan Temuan, hasil dan pembahasan dari pengambilan *main content* yang dilakukan pada penelitian ini.

4.1 Hasil Penelitian

Pada bagian berikut ini akan dijelaskan mengenai hasil yang didapatkan pada saat penelitian dilakukan dan pembahasan mengenai hasil tersebut.

4.1.1 Tahap *Preprocessing*

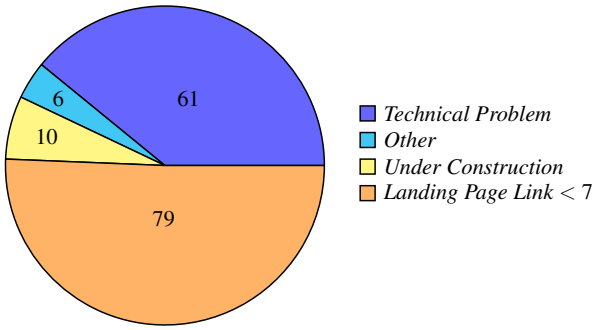
Langkah pertama sampai dengan langkah kelima pada tahap preprocessing adalah langkah yang dilakukan untuk mengambil *link url* yang valid dari situs web resmi pemerintah daerah di Indonesia, sebelum dilakukan pengecekan kategori setiap halaman web. *Link url* yang digunakan pada tahap satu atau tahap pengambilan *link url* (*crawling*) untuk situs web resmi pemerintah daerah di Indonesia adalah daftar *link url* yang sesuai pada situs Kementrian Komunikasi dan Informasi Indonesia dan sesuai dengan domain go.id.



Gambar 4.1: Grafik Pengambilan Link URL

Pengambilan *link url* dilakukan pada tanggal 13 Juni 2018. Selain itu proses pengambilan *link url* yang valid ini dilakukan dengan menggunakan waktu *timeout* sebesar 10 detik untuk setiap situs web resmi pemerintah daerah. Dengan total sebanyak 548 pemerintah daerah di Indonesia, jumlah pemerintah daerah yang berhasil dilakukan pengambilan *link url* yang valid adalah sebanyak 374 pemerintah

daerah dengan total *link url* sebanyak 31431 *link url* dan situs pemerintah daerah yang tidak berhasil dilakukan pengambilan *link url* yang valid adalah sebanyak 174 pemerintah daerah seperti yang terlihat pada gambar 4.1.



Gambar 4.2: Permasalahan Pengambilan Link URL

Dari 174 Pemerintah daerah yang tidak dapat dilakukan pengambilan *link url*, 18 diantaranya merupakan pemerintah daerah yang belum memiliki situs web resmi pemerintah daerah, dimana permasalahan yang dihadapi untuk 156 pemerintahan yang lain dapat dilihat pada gambar 4.2. Berdasarkan gambar 4.2, permasalahan yang timbul pada saat pengambilan *link url* adalah:

1. Halaman beranda atau *landing* yang berisi *link* ke *subdomain* atau *domain* lain yang umumnya disebut dengan halaman web portal dimana terdapat 79 situs web resmi pemerintah daerah yang termasuk kedalam kategori ini. Halaman web *landing* seperti ini hanya terdiri atas portal yang berisi *link* ke *subdomain* atau *domain* lain akan menyulitkan pengambilan konten yang mana umumnya *link* yang mengarah ke *subdomain* adalah *link* menuju ke aplikasi milik pemerintah daerah misalnya LPSE dimana akan menyulitkan dan mengurangi akurasi pengambilan konten. Selain itu halaman web semacam ini memiliki jumlah link valid (link yang mengarah pada *domain* yang sama dengan *url* resmi untuk setiap situs web resmi pemerintah daerah) kurang dari batas minimal jumlah *link* yang telah ditentukan yaitu minimal tujuh buah *link*.
2. Permasalahan teknis yang berasal dari *server*, yang mana sebagian besar permasalahan yang muncul pada kategori ini adalah situs yang memberikan HTTP *Status Code* 000 yang mengindikasikan bahwa *server* tidak memberikan

Tabel 4.1: Permasalahan Teknis pada Pengambilan Link url

Tipe Error	Jumlah
Encoding problem	1
Status 000	33
Status 500	2
Status 404	4
Status 403	2
Unconventional Redirection	15
iframe	3
Blank Page	2

respon yang valid terhadap *request* yang diberikan dalam proses pengambilan *link url* yang dapat diakibatkan situs web resmi pemerintah daerah tidak dapat diakses atau terjadi *timeout* pada saat pengambilan *link url*.

Selain permasalahan HTTP *Status Code* 000, juga terdapat berberapa permasalahan lain seperti HTTP *Status Code* 404 yang mengindikasikan bahwa *server* tidak menemukan halaman web yang diminta dimana pada kasus ini adalah halaman web landing dari situs web resmi pemerintah daerah di Indonesia. Secara lebih mendetail, permasalahan yang ditemui pada kategori permasalahan teknis pada saat pengambilan link url dapat dilihat pada tabel 4.1

Permasalahan teknis mengenai *unconventional url redirection* adalah permasalahan yang melakukan *redirect* ke halaman web lain dengan menggunakan metode yang tidak biasa seperti menggunakan metode ”http-equiv=refresh” seperti pada kode 4.1 dan menggunakan *script* seperti pada kode 4.2. Hal ini menyulitkan dalam melakukan pengambilan *link url* karena halaman web yang diterima hanya berupa kode seperti yang terlihat pada kode 4.1 dan kode 4.2. Permasalahan iframe adalah permasalahan dimana konten terenkapsulasi oleh iframe dan diambil pada tempat atau link lain sehingga menyulitkan pengambilan *link url* seperti pada kode 4.3.

Pada tabel 4.1, juga terlihat berberapa permasalahan teknis yang jarang ditemui, seperti permasalahan *unicode* dimana halaman web yang diberikan oleh *server* tidak menyertakan tipe *encoding* yang digunakan. Permasalahan se-

macam ini mungkin tidak terlihat ketika dilihat melalui *browser* karena pada umumnya *browser* akan menggunakan *encoding* UTF-8 ketika tipe *encoding* tidak dispesifikkan. Permasalahan lain yang cukup jarang terjadi adalah halaman web yang diberikan oleh server hanya berupa halaman web kosong tanpa adanya *tag* HTML atau *script* apapun sehingga tidak dapat dilakukan pengambilan *link url*.

3. Kategori *other* pada permasalahan pengambilan *link url* terdiri atas permasalahan non-teknis yang jarang terjadi seperti permasalahan mengenai *account suspended* yang mengindikasikan adanya permasalahan antara pemerintah daerah dengan pihak *server provider*, permasalahan SSL *expired* dimana merupakan permasalahan antara pihak pengelola situs web resmi pemerintah daerah dengan pihak SSL *provider* atau permasalahan website yang ter-*hack* oleh pihak yang tidak berkepentingan.

```
<meta http-equiv="refresh" content="0;URL='https://sukabumikab.go.id/portal/'" />
```

Kode 4.1: http-equiv refresh pada halaman web

```
<script>
window.location = "http://portal.belitungkab.go.id";
</script>
```

Kode 4.2: script redirection pada halaman web

```
<meta name="description" content="Website Pemerintah Kabupaten Luwu Timur">
  <frameset rows="*,1" framespacing="0" border="0" frameborder="NO">
  <frame src="http://www.luwutimurkab.go.id/lutim/" scrolling="auto" noresize>
  </frameset>
  <noframes>
<body>
</body>
```

Kode 4.3: iframe pada halaman web

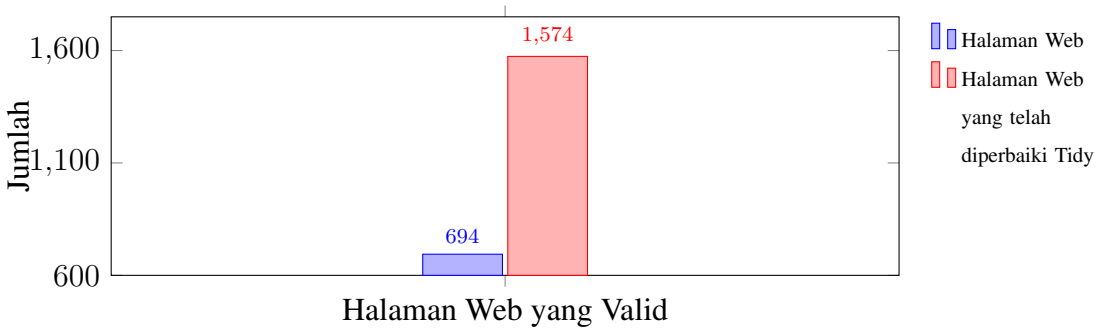
Langkah selanjutnya setelah didapatkan *link url* yang valid adalah melakukan pengecekan kategori halaman web untuk mengkategorisasikan halaman web sesuai dengan kategori Panduan Penyelenggaraan Situs Pemerintah Daerah. Pengecekan kategori dilakukan dengan mengacu kepada pengkategorian yang dilakukan oleh

Wisnu [Sugiyanto, 2017]. Pengecekan kategori pada halaman web juga dilakukan menggunakan pengaturan waktu *timeout* sebesar 10 detik. Setelah dilakukan pengecekan kategori setiap halaman web didapatkan sebanyak 3267 *link url* dari total 31431 *link url* yang sesuai dengan dengan kategori Panduan Penyelenggaraan Situs Pemerintah Daerah.

Tabel 4.2: Error Message yang dihasilkan W3C Validator

Error Type	Example	W3C Validator Message
Mismatched end tags	subheading	But there were open elements
Misnested tags	boldbold italic bold?	Violates nesting rules
Missing end tag / Unclosed Pairs	<h1>heading	Seen but an element of the same type was already open
Mixed-up tags	<p>new paragraph bold text <p>some more bold text	Not allowed on element
Missing "/" in end tags	<h1><hr>heading<h1>	Unclosed element
List markup with missing tags	<body> 1st list item 2nd list item	Unclosed element
Tags without terminating >	<h1><hr>heading</h1	Missing ">" immediately before
Typographical Error	<dov>heading</dov>	Not allowed as child of element
Structural Fatal Error	<html></html>	Cannot recover after last error

Langkah ke tujuh dari tahap preprocessing adalah tahap validasi atau pengecekan struktur HTML *tag* dari setiap halaman web pada ke 3267 *link url* tersebut. Proses validasi atau pengecekan struktur HTML *tag* pada halaman web dilakukan dengan menangkap *error message* yang dihasilkan pada API W3C *Validation service* menggunakan python *library* py_w3c dibandingkan dengan rule validasi struktur HTML tag pada tabel 3.1. Daftar *error message* yang digunakan dapat dilihat pada tabel 4.2.



Gambar 4.3: Hasil Validasi Halaman Web

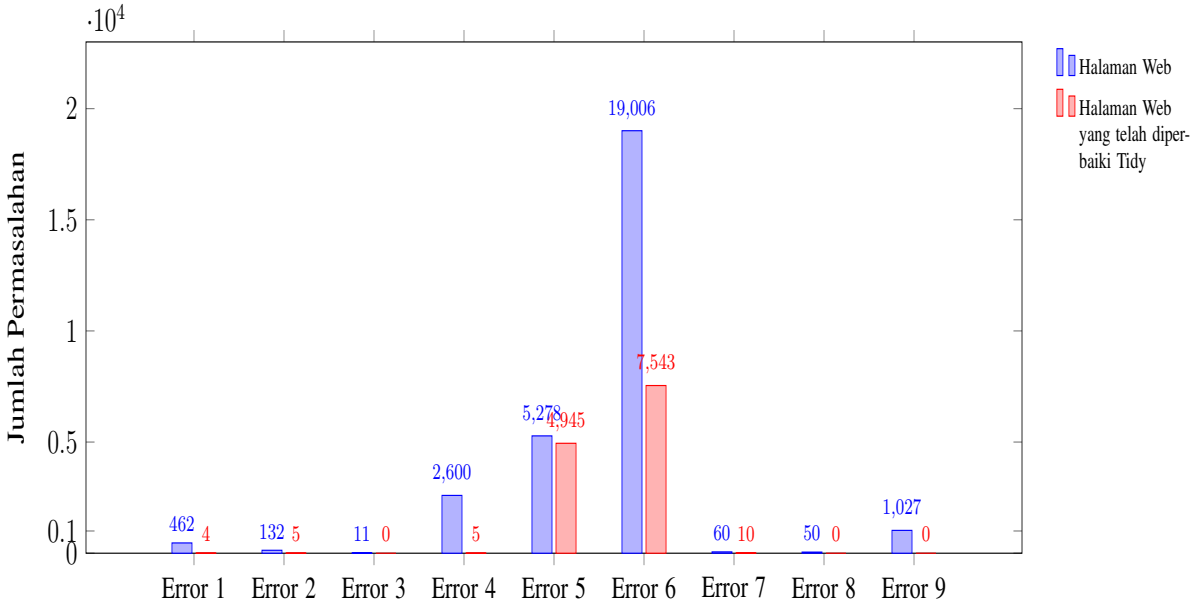
Pada penelitian ini, selain melakukan validasi atau pengecekan struktur HTML *tag* pada halaman web, juga dilakukan pengecekan pada halaman web yang telah dilakukan perbaikan melalui aplikasi tidy. Perbandingan dari hasil validasi atau pengecekan struktur HTML tag pada halaman web dan halaman web yang telah diperbaiki menggunakan tidy dapat dilihat pada gambar 4.3.

Untuk Halaman web, terdapat 694 halaman web yang lolos dalam validasi atau pengecekan struktur halaman web yang tersebar pada 83 situs pemerintah daerah. Sedangkan pada halaman web yang telah diperbaiki oleh aplikasi tidy terdapat 1574 halaman web yang lolos dalam validasi atau pengecekan struktur halaman web yang tersebar pada 174 situs pemerintah daerah. Lebih detail mengenai perbandingan jumlah *error* pada setiap *error type* pada halaman web dan halaman web yang telah diperbaiki oleh tidy dapat dilihat pada gambar 4.4. Terlihat bahwa jumlah *error type* terbanyak adalah error 6 yaitu mengenai permasalahan *mixed-up tags* dimana jumlah kejadian yang ditemui berjumlah sebanyak 19.006 kali kejadian.

Selain itu dengan melihat gambar 4.4 , terlihat bahwa dengan melakukan perbaikan pada halaman web dengan menggunakan aplikasi tidy dapat mengurangi jumlah *error* pada struktur HTML *tag* secara signifikan walaupun tidak dapat menghilangkan *error* pada struktur HTML tag secara keseluruhan.

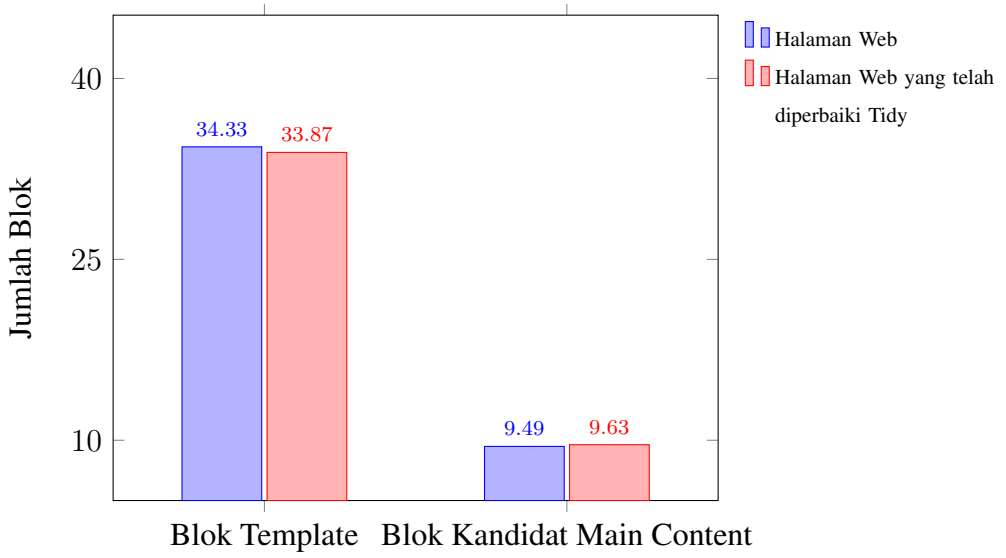
4.1.2 Tahap Pengambilan *Main Content* dengan Pendekatan *Template-Based*

Hasil halaman web yang telah melewati tahap *preprocessing*, yaitu 694 halaman web dan 1574 halaman web yang telah diperbaiki menggunakan aplikasi tidy, kemudian akan dilakukan pengambilan *main content* dengan menggunakan pendekatan *template-based*. Pada gambar 4.5, terlihat bahwa rata-rata jumlah blok *template* yang ditemukan pada halaman web adalah sebanyak 35 blok atau segmen dan pada tidy berjumlah 34 blok atau segmen sedangkan untuk blok kandidat *main content* pada sebuah halaman web ditemukan rata-rata 9.49 blok untuk halaman web dan 9.63 blok untuk halaman web yang telah diperbaiki dengan tidy. Blok atau segmen yang dimaksud disini adalah sebuah blok atau segmen yang berada pada sebuah



Gambar 4.4: Jumlah Error Validasi Halaman Web

halaman web yang bisa berupa satu buah node atau HTML *tag* atau lebih dari satu buah node atau HTML *tag*.



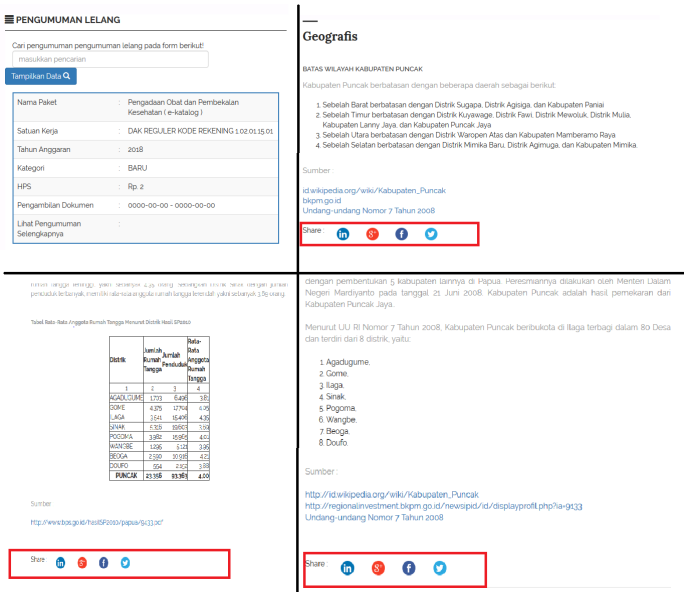
Gambar 4.5: Jumlah Rata-Rata Blok Yang Ditemukan Pada Halaman Web

Besarnya rata-rata blok unik yang ditemukan pada penelitian ini bisa disebabkan oleh berberapa permasalahan yang muncul dari temuan-temuan berikut ini:

1. Penelitian ini menggunakan jumlah batas kesamaan *node* atau *threshold* sebesar jumlah halaman web yang diproses pada saat pengambilan *main content*

dengan pendekatan *Template-Based*, namun ditemui bahwa pada satu situs web resmi pemerintah daerah dapat mempunyai layout yang sedikit berbeda untuk informasi yang ditampilkan , misalnya halaman web mengenai informasi sejarah akan mempunyai *layout* yang sedikit berbeda dengan *layout* halaman web mengenai informasi peraturan daerah. Meskipun demikian, sebagian besar halaman web antara kedua halaman web tersebut akan tetap memiliki kesamaan yang dapat diambil menggunakan pendekatan *template-based*.

Dengan adanya perbedaan kecil pada *layout* tersebut dapat mengurangi akurasi dari tahap pengambilan *main content* dengan menggunakan pendekatan *template-based*, dikarenakan dengan jumlah nilai *threshold* atau batas kesamaan *node* yang digunakan pada penelitian ini sebanyak jumlah halaman web yang diproses, sehingga apabila sebuah *node* tidak muncul pada satu halaman web, maka *node* tersebut dianggap sebagai sebuah *node* yang unik.



Gambar 4.6: Perbedaan *Layout* dalam Penyajian *Main Content* Pada Halaman Web

Sebagai contoh hal ini dapat dilihat seperti pada gambar 4.6, Dimana terdapat 4 halaman web yang lolos validasi akan diproses pada tahap pengambilan *main content* dengan pendekatan *template-based*. Terlihat terdapat blok yang

muncul pada 3 halaman web yang ditandai dengan kotak merah, akan tetapi kotak merah tersebut tidak muncul pada satu halaman web lain sehingga blok pada kotak merah tersebut dianggap sebagai sebuah blok yang unik. Bergantung terhadap struktur HTML dari halaman web yang ada permasalahan tersebut dapat menimbulkan 2 hasil dimana blok yang ditandai dengan kotak merah dianggap menjadi blok unik yang terpisah pada 3 halaman web tersebut dan menambah jumlah blok atau segmen yang bukan *main content* atau *parent* dari blok tersebut dianggap menjadi blok unik pada 4 halaman web tersebut yang mengakibatkan *feature set* dari blok *main content* berubah menyerupai blok yang bukan *main content*.

2. Pemerintah daerah menulis atau mem-publish informasi atau *main content* dengan menggunakan teknologi *embedded* seperti kode 4.6 dan iframe seperti kode 4.5.

   blok atau segmen seperti ini masih tetap dapat terambil pada saat pengambilan *main content* melalui pendekatan *template-based*, akan tetapi hal ini akan mengurangi akurasi dalam tahap selanjutnya yaitu pada tahap pengambilan *main content* dengan pendekatan klasifikasi *machine learning* karena *feature set* yang terambil dari blok atau segmen semacam ini akan sangat mirip dengan *feature set* dari blok atau segmen yang bukan *main content*.

3. Terdapat pemerintah daerah yang memiliki *Attribute Node* dari HTML *tag* yang berubah secara dinamis. Seperti yang terlihat pada perbandingan blok atau segmen yang terambil dari dua halaman web yang dapat dilihat pada kode 4.7 dan kode 4.8.

   Jika dilihat pada kode 4.7 dan 4.8, untuk baris 35 sampai dengan baris 37 merupakan blok atau segmen yang seharusnya diambil pada tahap pengambilan *main content* dengan menggunakan pendekatan *template-based*, akan tetapi blok atau segmen yang terambil adalah ditunjukkan seperti pada kode 4.7 dan kode 4.8 , dimana masih terdapat kesamaan blok atau segmen yang terlihat pada baris ke 38 dan seterusnya yang merupakan *sidebar* dari kedua halaman web tersebut yang seharusnya dapat dihilangkan pada saat meng-

gunakan pendekatan *template-based*. Kejadian ini disebabkan karena pada *Node* yang terdapat pada baris ke-1 terjadi sedikit perbedaan pada nama class yaitu "post-254" pada halaman web satu seperti pada kode 4.7 dan "post 268" pada halaman web dua seperti pada kode 4.8.

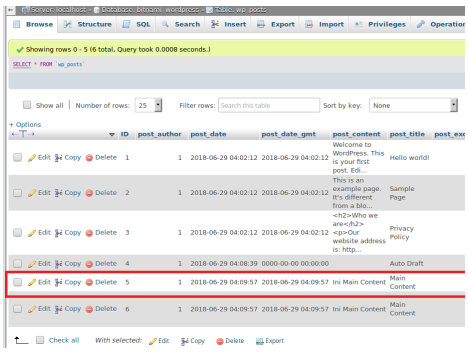
Dengan perbedaan kecil tersebut maka *node* tersebut pada masing-masing halaman mempunyai *Attribute Node* yang berbeda sehingga dianggap sebagai blok yang unik pada masing-masing halaman. Kejadian seperti ini bisa disebabkan karena situs pemerintah daerah menggunakan *content management system* (CMS) yang berperilaku seperti demikian. hal seperti ini akan dapat mengganggu pada akurasi pada tahap pengambilan *main content* dengan klasifikasi *machine learning* karena *feature set* yang terambil dari blok atau segmen semacam ini akan sangat mirip dengan feature set dari blok atau segmen yang bukan *main content*.

Untuk memahami kenapa permasalahan ini muncul pada situs yang menggunakan *Content Management System*, perlu dipahami terlebih dahulu tentang bagaimana sebuah *Content Management System* menyimpan *main content*. Secara umum, penyimpan *main content* pada *Content Management System* didasarkan atas arah atau tujuan yang ingin dicapai dari *Content Management System* tersebut yaitu apakah *Content Management System* tersebut ingin mengarah ke tipe *dynamic site* atau *static site*. *Content Management System* yang mengarah ke tipe *dynamic site* umumnya menyimpan *main content* ke dalam sebuah *database*, sedangkan *Content Management System* yang mengarah ke tipe *static site* akan menyimpan *main content* pada sebuah *file* HTML. Salah satu keuntungan pada *dynamic site* adalah kemampuannya menyimpan *main content* ke dalam sebuah *database* atau tempat dan menggunakan atau memperlihatkan *main content* tersebut ke berberapa halaman web yang berbeda. Sedangkan salah satu keuntungan pada *static site* adalah kecepatan dalam melakukan *load* sebuah halaman web karena hanya perlu menampilkan sebuah *file* HTML.

Permasalahan mengenai *Attribute Node* dari HTML tag yang berubah secara

dinamis yang ditemukan pada penelitian ini umumnya ditemukan pada *Content Management System* yang mempunyai tipe *dynamic site*. Kemungkinan besar permasalahan ini muncul terkait dengan cara pengambilan *main content* dari *database Content Management System* untuk ditampilkan pada sebuah halaman web.

Sebagai contoh pada kode 4.8 terlihat bahwa terdapat kata “post-254” di dalam *tag* HTML pada baris pertama, dimana kemungkinan besar kata “post-254” menunjukkan identitas dari *main content* tersebut di dalam database *Content Management System*. Sebagai contoh, pada *Content Management System* Wordpress akan di-*publish* sebuah halaman web yang berisi kata ”Main Content”. Terlihat pada gambar 4.7, bahwa wordpress menyimpan kata ”Main Content” pada *database* dengan id sama dengan 5 seperi yang ditandai dengan kotak merah pada gambar 4.7. Jika dilihat pada *source-page* pada halaman web maka akan terdapat kata ”post-5” seperti pada kode 4.4 yang dapat diartikan bahwa pada halaman web tersebut mengambil *main content* pada *database wordpress* dengan id sama dengan 5.



Gambar 4.7: Penyimpanan Main Content di Database Pada *Content Management System* Wordpress

```
1        <article id="post-5" class="post-5 post type-post status-publish format-standard hentry category-
             ↪ uncategorized">
2            <div class="entry-content">
3                <p>Ini Main Content</p>
4            </div>
5        </article>
```

Kode 4.4: HTML Source Pada Halaman Web yang Dipublish pada *Content Management System* Wordpress

Hal ini menunjukkan bahwa *tag-tag* dinamis yang ditemui pada penelitian lebih mengarah tentang bagaimana sebuah *Content Management System* menyimpan dan menampilkan *main content* pada halaman web. Dengan demikian, permasalahan mengenai *Attribute Node* dari HTML tag yang berubah secara dinamis ini akan memiliki pengaruh yang berbeda antara satu *Content Management System* dengan *Content Management System* lain tergantung bagaimana *Content Management System* tersebut menyimpan dan menampilkan *main content*.

Untuk penelitian kedepan, salah satu metode yang dapat dilakukan untuk mengatasi permasalahan seperti ini adalah dengan menggunakan algoritma *text similarity* untuk membandingkan antara kedua *Attribute Node* pada masing-masing halaman web dan apabila tingkat kesamaan antara kedua *Attribute Node* melebihi dari batas yang ditentukan maka kedua *Attribute Node* tersebut dianggap sama.

```
<article class="content-page post-1059 page type-page status-publish hentry" id="post-1059">
 <header class="page-header">
  <h1 class="page-title entry-title s-font-size-title font-inherit">
   Wisata
  </h1>
 </header>
 <div class="entry-content clearfix">
  <iframe align="center" frameborder="yes" height="760px" name="frame1" scrolling="auto" src="http://
      ↪ pariwisatakotabarupulaulaut.blogspot.co.id/" style="border: 1px solid;" width="100%">
  </iframe>
 </div>
</article>
```

Kode 4.5: Contoh *main content* pada iframe pada halaman web

```
<article class="post-3094 page type-page status-publish hentry" id="post-3094">
 <div class="entry-content">
  <p style="text-align: justify;">
   <a class="pdfemb-viewer" data-height="max" data-toolbar="bottom" data-toolbar-fixed="on"  data-width="
       ↪ max" href="http://bondowosokab.go.id/v2/wp-content/uploads/2014/06/LAPORAN-BENCANA-ALAM-2015.
       ↪ pdf" style="">
    LAPORAN-BENCANA-ALAM-2015
    <br/>
   </a>
  </p>
 </div>
</article>
```

Kode 4.6: Contoh teknologi *embedded* pada halaman Web

```
<div class="post-268 page type-page
    ↪ status-publish hentry" id="
    ↪ core">
 <div class="c-container">
  <div class="row">
  <div class="col-md-6 middle-column
      ↪ col-md-push-3">
    <div class=" m-has-breadcrumbs"
        ↪ id="page-header">
     <div class="page-title">
      <h1>
       Lambang Daerah
      </h1>
     </div>
     <div class="breadcrumbs">
      <ul>
      <li class="home">
        <a href="http://pacitankab.
            ↪ go.id">
         Home
        </a>
      </li>
      <li>
        Lambang Daerah
      </li>
      </ul>
     </div>
    </div>
    <div id="page-content">
      Main Content
    </div>
    <hr class="c-separator m-margin-
        ↪ top-small m-margin-
        ↪ bottom-small m-
        ↪ transparent hidden-lg
        ↪ hidden-md"/>
  </div>
  <div class="col-md-3 left-column
      ↪ col-md-pull-6">
    <div class="side-menu m-left-
        ↪ side m-show-submenu">
      SIDEBAR
```

Kode 4.7: Tag HTML Dinamis pada Halaman Web

```
<div class="post-254 page type-page
    ↪ status-publish hentry" id="
    ↪ core">
 <div class="c-container">
  <div class="row">
  <div class="col-md-6 middle-column
      ↪ col-md-push-3">
    <div class=" m-has-breadcrumbs"
        ↪ id="page-header">
     <div class="page-title">
      <h1>
       Geografis
      </h1>
     </div>
     <div class="breadcrumbs">
      <ul>
      <li class="home">
        <a href="http://pacitankab.
            ↪ go.id">
         Home
        </a>
      </li>
      <li>
        Geografis
      </li>
      </ul>
     </div>
    </div>
    <div id="page-content">
      Main Content
    </div>
    <hr class="c-separator m-margin-
        ↪ top-small m-margin-
        ↪ bottom-small m-
        ↪ transparent hidden-lg
        ↪ hidden-md"/>
  </div>
  <div class="col-md-3 left-column
      ↪ col-md-pull-6">
    <div class="side-menu m-left-
        ↪ side m-show-submenu">
      SIDEBAR
```

Kode 4.8: Tag HTML Dinamis pada Halaman Web

4.1.3 Tahap Pengambilan Main Content dengan pendekatan Klasifikasi *Machine Learning*

Setelah didapatkan blok yang merupakan kandidat *main content* dari tahap pengambilan *main content* dengan menggunakan pendekatan *template-based* kemudian dilakukan pendekatan klasifikasi *machine learning* untuk menentukan sebuah blok

Tabel 4.3: Statistik Data

	Halaman Web				Halaman Web yang telah diperbaiki Tidy			
	Words	Sentence	Links	Text Density	Words	Sentence	Links	Text Density
AVERAGE	53,81	2,50	6,88	65,35	463,36	32,75	10,85	83,25
MAX	15391	1585	205	1152,31	14980	1585	242	3057
MIN	16	0	0	0,62	10	0	0	1
MEDIAN	278	14	3	28,33	306	16	1	36
QUARTILE 1	146	5	0	10,475	165,00	7,00	0,00	13,35
QUARTILE 3	524,50	32,00	12,00	73,09	523,00	36,00	7,00	85,00
STDEV	1144,12	101,52	20,48	110,15	734,84	72,56	25,70	163,20
Jumlah Main Content	479				1037			

atau tag sebagai *main content* atau bukan *main content*.

4.1.3.1 Pembentukan Model Klasifikasi

Tahap klasifikasi *machine learning* dilakukan dengan menggunakan Naïve Bayes *Classifier* dan menggunakan 4 *feature set* yang telah dimodifikasi dari Yao [Yao and Zuo, 2013]. Dari tahap pengambilan *main content*dengan menggunakan pendekatan *template-based* didapatkan 5496 Dataset untuk halaman web dan 13319 dataset untuk halaman web yang diperbaiki dengan menggunakan tidy. Kemudian untuk setiap data pada dataset dilakukan pelabelan menjadi *main content* atau bukan *main content*. Berdasarkan hasil dataset yang telah diberikan label, hasil statistik untuk data yang termasuk main content pada dataset halaman web dan halaman web yang telah diperbaiki Tidy dapat dilihat pada tabel 4.3.

Efek dari temuan-temuan yang dijelaskan pada tahap pengambilan *main content* dengan pendekatan *template-based* sebelumnya dapat terlihat dengan melihat nilai pada kolom MAX dan MIN. Contohnya *main content* yang menggunakan teknologi *embeded* akan muncul dengan nilai *words*, *sentence, links* dan *text density* yang sangat kecil karena isi dari halaman web tidak dapat terambil sepenuhnya. Di lain pihak, *main content* yang memiliki *Attribute Node* dari HTML *tag* yang berubah secara dinamis akan memiliki nilai *words, sentences, link* dan *text density* yang besar karena pada tahap pengambilan *main content* dengan *template-based* tidak dapat dilakukan pengambilan blok atau segmen secara sempurna.

Dengan melihat kolom rata-rata (*average*) pada tabel 4.3, Suatu blok atau segmen pada halaman web memiliki kemungkinan besar sebagai *main content* apabi-

la blok atau segmen tersebut memiliki setidaknya 54 kata dan terdiri atas 3 kalimat atau lebih dengan jumlah link kurang dari 7 dan kepadatan teks (*text density*) berkisar pada nilai 66. Sedangkan, suatu blok atau segmen pada halaman web yang telah diperbaiki oleh tidy memiliki kemungkinan besar sebagai *main content* apabila blok atau segmen tersebut memiliki setidaknya 466 kata dan terdiri atas 33 kalimat atau lebih dengan jumlah link kurang dari 11 dan kepadatan teks (text density) berkisar pada nilai 84. Akan tetapi nilai-nilai yang muncul ini akan bersifat kurang representatif mengingat berberapa permasalahan yang muncul berdasarkan hasil temuan-temuan yang telah dijelaskan sebelumnya. Hal ini juga berlaku terhadap muncul terhadap nilai standar deviasi yang muncul pada tabel 4.3.

Meskipun demikian, nilai yang ada pada kolom *Quartile* 1, median dan *Quartile 3* pada tabel 4.3 dapat memberikan gambaran lebih jelas tentang bagaimana *main content* yang ada pada situs web resmi pemerintah daerah di Indonesia. Sebuah *main content* pada halaman web pada situs web resmi pemerintah daerah di Indonesia akan memiliki jumlah kata antara 146 hingga 525 kata dengan jumlah kalimat antara 5 hinggga 32 kalimat. Serta *main content* tersebut memiliki jumlah link kurang dari 12 link dengan text density antara 11 hingga 73. Sedangkan untuk halaman web yang telah diperbaiki oleh tidy akan memiliki jumlah kata antara 165 hingga 523 kata dengan jumlah kalimat antara 7 hinggga 36 kalimat. Serta main content tersebut memiliki jumlah link kurang dari 7 link dengan text density antara 14 hingga 85.

Dataset yang telah dibangun kemudian dibagi menjadi dua bagian yaitu 70% dan 30%, dimana 70% dari *dataset* akan digunakan untuk tahap *training data* untuk klasifikasi *machine learning* dan 30% dari *dataset* akan digunakan sebagai tahap *testing classifier* untuk klasifikasi *machine learning*. Pada langkah *training* akan dilakukan evaluasi dengan menggunakan *cross validation* K-Fold dengan jumlah *split* sebesar 5 *split*. Hasil dari evaluasi *Cross validation* dapat dilihat pada tabel 4.4 untuk halaman web dan tabel 4.5 untuk halaman web yang telah diperbaiki oleh tidy.

Tabel 4.4: Hasil Evaluasi Model Naïve Bayes Halaman Web

Split	Label	Precision	Recall	F1-Score
1	Main Content	0.75	0.36	0.48
	Bukan Main Content	0.94	0.99	0.96
	Avg / Total	0.93	0.93	0.92
2	Main Content	0.7	0.32	0.44
	Bukan Main Content	0.94	0.99	0.97
	Avg / Total	0.93	0.94	0.92
3	Main Content	0.73	0.36	0.48
	Bukan Main Content	0.95	0.99	0.97
	Avg / Total	0.93	0.94	0.93
4	Main Content	0.64	0.28	0.39
	Bukan Main Content	0.94	0.99	0.96
	Avg / Total	0.91	0.93	0.91
5	Main Content	0.76	0.45	0.56
	Bukan Main Content	0.94	0.98	0.96
	Avg / Total	0.92	0.93	0.92
	Avg Accuracy	0.929778934		

Tabel 4.5: Hasil Evaluasi Model Naïve Bayes Halaman Web yang telah diperbaiki Tidy

Split	Label	Precision	Recall	F1-Score
1	Main Content	0.75	0.42	0.53
	Bukan Main Content	0.96	0.99	0.97
	Avg / Total	0.94	0.95	0.94
2	Main Content	0.82	0.38	0.51
	Bukan Main Content	0.94	0.99	0.97
	Avg / Total	0.93	0.94	0.93
3	Main Content	0.78	0.45	0.57
	Bukan Main Content	0.95	0.99	0.97
	Avg / Total	0.94	0.95	0.94

Tabel 4.5: Hasil Evaluasi Model Naïve Bayes Halaman Web yang telah diperbaiki Tidy

Split	Label	Precision	Recall	F1-Score
4	Main Content	0.75	0.41	0.53
	Bukan Main Content	0.96	0.99	0.97
	Avg / Total	0.95	0.95	0.95
5	Main Content	0.76	0.33	0.46
	Bukan Main Content	0.94	0.99	0.97
	Avg / Total	0.93	0.94	0.93
	Avg Accuracy	0.936695279		

Seperti yang telah dibahas sebelumnya, Temuan-temuan yang dijelaskan pada tahap pengambilan *main content* dengan pendekatan *template-based* sangat berpengaruh terhadap hasil yang muncul pada tabel 4.4 dan table 4.5. Pada tabel 4.4 dan tabel 4.5 nilai untuk label "*main content*" untuk *precision*, *recall* dan *f1-score* memiliki nilai cukup rendah. Dengan nilai *recall* yang kurang dari 0.5 dan *precision* berada di kisaran nilai 0.75, menandakan bahwa model yang telah dibuat dapat memprediksi blok yang merupakan *main content* dengan cukup baik walaupun tidak dapat mengambil semua blok yang merupakan *main content*. Hal ini bisa diakibatkan karena temuan-temuan yang dibahas pada tahap pengambilan *main content* dengan pendekatan *template-based* yaitu:

1. Sebuah blok yang seharusnya dapat diproses lebih mendalam atau merupakan *template*, dianggap unik karena blok atau segmen tersebut tidak muncul pada satu halaman web. Contohnya seperti yang terlihat pada gambar 4.6 dimana sebuah bagian yang ditandai dengan kotak merah muncul pada tiga halaman web dan tidak muncul pada satu halaman web. Sehingga *node* tersebut dianggap sebagai sebuah bagian yang unik pada sebuah blok dimana dapat mengakibatkan *feature set* yang terbentuk untuk blok *main content* akan menyerupai blok yang bukan *main content*. Permasalahan ini muncul pada blok yang merupakan *main content* sebanyak 17% pada halaman web dan 25% pada halaman web yang telah diperbaiki oleh Tidy.

2. Blok dengan *main content* yang menggunakan teknologi *embedded* atau iframe muncul sebanyak 2% pada halaman web dan 1% pada halaman web yang telah diperbaiki tidy.
3. Blok dengan *main content* yang memiliki *Attribute Node* dari HTML tag yang berubah secara dinamis muncul sebanyak 30% pada halaman web dan 16% pada halaman web yang telah diperbaiki tidy.

Permasalahan-permasalahan yang muncul tersebut dapat membuat *feature set* yang dimiliki oleh blok *main content* berubah dan akan sangat mirip dengan *feature set* yang dimiliki oleh label "bukan *main content*". Berbeda dengan label "*main content*", nilai yang didapatkan pada *precision*, *recall* dan *f1-score* untuk label "bukan *main content*" adalah sangat baik. Hal ini dapat diartikan bahwa model yang telah dibangun ini dapat mengindentifikasi blok yang bukan *main content* dengan sangat baik.

4.1.3.2 Improvisasi Feature Set

Dengan adanya berberapa permasalahan yang memberikan hasil yang buruk pada model yang telah dibuat seperti yang terlihat pada tabel 4.4 dan tabel 4.4, maka pada penelitian dicoba untuk meningkatkan hasil yang didapat dengan mengubah atau memperbaiki *feature set* yang digunakan terutama untuk meningkatkan nilai *precision* dan *recall* pada label *main content*.

Hal pertama yang dilakukan adalah melihat apakah terdapat fitur yang redundan ketika dilakukan pembangunan model. Hal tersebut dapat dilakukan dengan melihat nilai pearson *correlation matrix* dari setiap fitur. Pada tabel 4.6 adalah tabel pearson *correlation matrix* pada 4 *feature set* yang digunakan pada pembentukan model klasifikasi.

Tabel 4.6: Hasil *Correlation Matrix* Untuk 4 Fitur

	Jumlah Kata	TD	Jumlah Kalimat	Jumlah link
Jumlah Kata	1	0.608182	0.890493	0.418554
TD	0.608182	1	0.460258	0.442043
Jumlah Kalimat	0.890493	0.460258	1	0.111669
Jumlah link	0.418554	0.442043	0.111669	1

Dengan melihat tabel 4.6, terlihat bahwa fitur jumlah kata dan jumlah kalimat memiliki hubungan yang cukup kuat dengan nilai pearson *correlation* sebesar 0.890493. Walaupun demikian, nilai pearson *correlation* tersebut masih belum cukup tinggi atau kurang dari 0.95 (dengan tingkat signifikansi 0.05) untuk menandakan bahwa kedua fitur tersebut redundan. Sehingga didapatkan hasil bahwa tidak terdapat fitur yang redundan untuk ke empat fitur tersebut. Dengan tidak adanya fitur yang redundan, kemudian akan dilihat apakah terdapat fitur yang tidak signifikan atau tidak relevan dengan melakukan pengecekan terhadap nilai t-value dan nilai p-value dari pengetesan Chi-Square Test of Independence.

Nilai dari p-value dan t-value untuk setiap fitur dapat dlihat pada tabel 4.7. Pada tabel 4.7 terdapat berberapa fitur yang memiliki nilai p-value sebesar 2.22e-308, walaupun sebenarnya nilai p-value yang dihasilkan lebih kecil dari 2.22e-308. Hal ini disebabkan karena perhitungan p-value pada library stats pada python menggunakan tipe *float* dimana rentang minimum suatu nilai adalah sebesar 2.22e-308. Sehingga jika suatu nilai memiliki nilai lebih kecil dari 2.22e-308 maka nilai tersebut akan dibulatkan menjadi 0.0.

Dari tabel 4.6 terlihat bahwa setiap fitur memiliki nilai p-value yang kurang dari 0.05 (dengan tingkat signifikansi 0.05) sehingga dapat dikatakan bahwa setiap fitur merupakan fitur yang relevan dalam pembentukan model. Sedangkan untuk nilai t-value pada 4.7 terlihat bahwa nilai untuk t-value pada setiap fitur memiliki nilai diatas 7, dimana apabila menggunakan tingkat signifikansi sebesar 0.05 dan critical value sebesar 1.812 maka terlihat bahwa setiap fitur merupakan fitur yang relevan dalam pengambilan *main content*.

Tabel 4.7: Hasil Chi-Square Test of Independence Untuk 4 Fitur

	T-Value	P-Value
Words	12.079	2.22e-308
Sentence	7.638	2.22e-308
Link	16.623	5.9e-148
TD	19.425	2.22e-308

Setelah melihat hasil yang muncul pada tabel 4.6 dan tabel 4.7, maka diusulkan berberapa tambahan fitur untuk meningkatkan akurasi dari model yang terbentuk.

Fitur tambahan tersebut adalah:

```
<div>
    <p> ini contoh kalimat main content </p>
    </br>
    <p> ini adalah main content </p>
    <table>
            <tr>
                    <td><bold>content</bold></td>
                    <td>Halaman</td>
                    <td>Web</td>
            </tr>
    </table>
    <ul>
            <li><italic>content</italic></li>
            <li>HTML</li>
            <li>Situs</li>
    </ul>
</div>
```

Kode 4.9: Contoh Untuk *Feature Set* Tambahan

1. *Maximum Consecutive Word*

   Maximum consecutive word merupakan nilai tertinggi pada kata yang muncul secara konsekutif diantara tag HTML pada blok atau segmen. Fitur ini merupakan fitur yang dimodifikasi dari penelitian yang dilakukan oleh Kohlscutter [Kohlschütter et al., 2010] dimana dia menggunakan jumlah kata yang berada di antara *tag* HTML. Sebagai contoh pada kode 4.9, akan menghasilkan nilai tertinggi kata berurutan (*consecutive*) sebesar 6 yang didapat dari tag ” ini contoh kalimat pada main content ”.

2. *Mean Consecutive Word*

   Mean Consecutive Word merupakan rata-rata nilai pada kata yang muncul secara konsekutif diantara tag HTML pada blok atau segmen. Fitur ini merupakan fitur yang dimodifikasi dari penelitian yang dilakukan oleh Kohlscutter [Kohlschütter et al., 2010] dimana dia menggunakan jumlah kata yang berada di antara *tag* HTML misalnya diantara tag dan . Sebagai contoh pada kode 4.9, akan memberikan perhitungan nilai rata-rata kata berurutan (*consecutive*) seperti : 6 (ini contoh kalimat main content) + 4 (ini adalah main content) + 1 (Kopi) + 1 (Teh) + 1 (Susu) + 1 (Kopi) + 1 (Teh) + 1 (Susu) yang menghasilkan nilai 16. Kemudian hasil tersebut dibagi dengan jumlah kemunculan kata berurutan yaitu sebanyak 8 maka menghasilkan nilai *mean*

consecutive words sebesar 2.

3. *Max Occurence Word*

   Max Occurence Word merupakan jumlah kemunculan dari kata (*case insensitive*) yang paling banyak muncul pada blok atau segmen. Secara umum sebuah teks atau paragraf akan memiliki sejumlah kata yang muncul berulang kali. Kata-kata tersebut umumnya merupakan kata mengenai topik yang dibahas pada teks atau paragraph tersebut seperti kata mengenai pemerintah daerah pada halaman web yang membahas mengenai sejarah atau selayang pandang. Selain itu kata-kata tersebut juga memiliki kemungkinan sebagai sebuah kata hubung atau konjungsi seperti kata “dan” dan kata “atau” yang secara linguistik dapat mengindikasikan terdapat sebuah kalimat atau paragraph. Sebagai contoh pada kode 4.9, akan menghasilkan nilai *max occurence word* sebesar 4 yang dihasilkan dari kata ”content” yang muncul 4 kali.

4. *Text Formatting*

   Text formatting merupakan jumlah *tag* HTML mengenai *text formatting* yang ada pada blok atau segmen. Fitur ini digunakan untuk menghitung jumlah *tag* yang berhubungan dengan *text formatting* pada sebuah blok atau segmen seperti *tag* <bold> atau tag <italic>. *tag* yang termasuk kedalam *text formatting* mengacu kepada organisasi w3 [w3tech, 2018a] seperti yang terlihat pada tabel 4.8. Sebagai contoh pada kode 4.9, akan menghasilkan nilai *text formatting* sebesar 2 yang dihasilkan dari 1 *tag* <bold> dan 1 *tag* <italic>.

Tabel 4.8: *Tag* HTML untuk *Text Formatting*

Tag	Definition
<acronym>	Defines an acronym
<abbr>	Defines an abbreviation or an acronym
<address>	Defines contact information for the author/owner of a document/article
<b>	Defines bold text
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it

Tag	Definition
<bdo>	Overrides the current text direction
<big>	Defines big text
<blockquote>	Defines a section that is quoted from another source
	Defines centered text
<cite>	Defines the title of a work
<code>	Defines a piece of computer code
<del>	Defines text that has been deleted from a document
<dfn>	Represents the defining instance of a term
	Defines emphasized text
<font>	Defines font, color, and size for text
<i>	Defines a part of text in an alternate voice or mood
<ins>	Defines a text that has been inserted into a document
<kbd>	Defines keyboard input
<mark>	Defines marked/highlighted text
<meter>	Defines a scalar measurement within a known range (a gauge)
<pre>	Defines preformatted text
<progress>	Represents the progress of a task
<q>	Defines a short quotation
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<s>	Defines text that is no longer correct
<samp>	Defines sample output from a computer program
<small>	Defines smaller text
<strike>	Defines strikethrough text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<template>	Defines a template

Tag	Definition
<time>	Defines a date/time
<tt>	Defines teletype text
<u>	Defines text that should be stylistically different from normal text
<var>	Defines a variable
<wbr>	Defines a possible line-break

5. *Table formatting*

   Table formatting merupakan jumlah *tag* HTML mengenai *table formatting* yang ada pada blok atau segmen. Fitur ini digunakan untuk menghitung jumlah *tag* yang berhubungan dengan *table formatting* pada sebuah blok atau segmen seperti *tag* <tr> atau *tag* <td>. *tag* yang termasuk kedalam *table formatting* mengacu kepada organisasi w3 [w3tech, 2018a] seperti yang terlihat pada tabel 4.10. Sebagai contoh pada kode 4.9, akan menghasilkan nilai *table formatting* sebesar 5 yang dihasilkan dari 1 *tag* <table>, 1 *tag* <tr> dan 3 *tag* <td>.

Tabel 4.10: *Tag* HTML untuk *Table Formatting*

Tag	Definition
	Defines an unordered list
	Defines an ordered list
	Defines a list item
<dir>	Defines a directory list
<dl>	Defines a description list
<dt>	Defines a term/name in a description list
<dd>	Defines a description of a term/name in a description list

6. *Paragraph formatting*

   Paragraph formatting merupakan jumlah *tag* HTML mengenai *paragraph formatting* yang ada pada blok atau segmen. Fitur ini digunakan untuk menghitung jumlah *tag* yang berhubungan dengan *Paragraph formatting* pada sebuah blok atau segmen seperti *tag* <pr> atau *tag*
. *tag* yang termasuk kedalam *paragraph formatting* mengacu kepada organisasi w3 [w3tech, 2018a] seperti yang terlihat pada tabel 4.11. Sebagai contoh pada kode 4.9, akan menghasilkan nilai *paragraph formatting*

sebesar 3 yang dihasilkan dari 2 *tag* <p> dan 1 *tag*
.

Tabel 4.11: *Tag* HTML untuk *Paragraph Formatting*

Tag	Definition
<p>	Defines a paragraph
 	Inserts a single line break

7. *List formatting*

   List formatting merupakan jumlah *tag* HTML mengenai *list formatting* yang ada pada blok atau segmen. Fitur ini digunakan untuk menghitung jumlah *tag* yang berhubungan dengan *list formatting* pada sebuah blok atau segmen seperti *tag* atau *tag* . *tag* yang termasuk kedalam *list formatting* mengacu kepada organisasi w3 [w3tech, 2018a] seperti yang terlihat pada tabel 4.12. Sebagai contoh pada kode 4.9, akan menghasilkan nilai *list formatting* sebesar 4 yang dihasilkan dari 1 *tag* dan 3 *tag* .

Tabel 4.12: *Tag* HTML untuk *List Formatting*

Tag	Definition
<table>	Defines a table
<caption>	Defines a table caption
<th>	Defines a header cell in a table
<tr>	Defines a row in a table
<td>	Defines a cell in a table
<thead>	Groups the header content in a table
<tbody>	Groups the body content in a table
<col>	Specifies column properties for each column within a <colgroup>element
<colgroup>	Specifies a group of one or more columns in a table for formatting

Untuk melihat apakah terdapat fitur yang redundan dari penambahan fitur, maka dilakukan uji pearson *correlation matrix* dengan hasil yang dapat dilihat pada tabel 4.15. Pada tabel 4.15 terlihat bahwa korelasi paling tinggi dimiliki oleh fitur *list formatting* dengan fitur jumlah *link* dimana memiliki nilai sebesar 0.929632. Dengan Hasil tersebut, apabila menggunakan tingkat signifikasi sebesar 0.05 maka tidak terdapat fitur yang redundan.

Pada tabel 4.16 terlihat nilai untuk t-value pada setiap fitur, dimana apabila menggunakan tingkat signifikansi sebesar 0.05 dan critical value sebesar 1.812 maka terlihat bahwa setiap fitur merupakan fitur yang relevan dalam pengambilan *main content*. Sedangkan hasil dari Chi-Square Test of Independence seperti yang terlihat pada tabel 4.16 menunjukkan bahwa semua fitur memiliki nilai p-value untuk semua fitur bernilai kurang dari 0.05 sehingga semua fitur merupakan fitur yang relevan dalam pembentukan model.

Lebih Lanjut dengan melakukan *Confirmatory Factor Analysis* pada 11 fitur tersebut didapatkan nilai loading factor untuk masing-masing fitur yang terlihat seperti pada tabel 4.13. *Loading Factor* adalah *korelasi* antara *independent variable* (fitur) dan *dependent variable* (variabel prediksi *main content* atau bukan *main content*) dimana jika loading factor memiliki nilai lebih dari 0.4 maka fitur tersebut dapat diterima [Fornell and Larcker, 1981, Hair JF and W, 1998, JP, 1992, AL and HB, 1992] .

Pada tabel 4.13 terlihat bahwa nilai *loading factor* yang dimiliki pada 11 fitur lebih dari 0.4 dimana merupakan *loading factor* yang dapat diterima. Hal ini menunjukkan bahwa setiap variabel *independent* (fitur) memiliki korelasi yang signifikan dengan variabel *dependent* (variable prediksi *main content* atau bukan *main content*).

Tabel 4.13: Hasil Loading Factor dari CFA Untuk 11 Fitur

Fitur	Loading Factor
words	0.99944371
sentence	0.89081337
link	0.417745
TD	0.60870926
Maximum Consecutive Words	0.46597103
Mean Consecutive Words	0.4231396
Maximum Occurrence	0.90312309
Text Formatting	0.44714121
Table Formatting	0.63551313
Paragraph Formatting	0.61323819
List Formatting	0.45041003

Peningkatan fitur menjadi 11 fitur ini juga berdampak pada reliabilitas yang

didapat dimana terjadi peningkatan nilai cronbach alpha dari sebelumnya 0.793 ketika menggunakan 4 fitur menjadi 0.863 ketika menggunakan 11 fitur seperti yang terlihat pada tabel 4.14

Tabel 4.14: Komparasi Cronbach Alpha

Fitur	Cronbach Alpha
4 *Feature Set*	0.793
11 *Feature Set*	0.863

Kemudian dilakukan pembangunan model klasifikasi dengan pembagian 70% untuk *training* model dan 30% untuk validasi model dari total dataset yang dipergunakan untuk total keseluruhan tahap *training*. Pada gambar 4.9 dan gambar 4.10 terlihat mengenai grafik *sensitivity* dan *specificity* untuk label *main content* dan label bukan *main content* dari setiap *probability threshold* pada halaman web dan halaman web yang telah diperbaiki Tidy. Secara umum *probability threshold default* yang digunakan pada klasifikasi naive-bayes adalah 0.5 (yang ditandai dengan garis merah seperti pada gambar 4.9) dimana jika suatu blok atau segmen memiliki *probability* 0.41 pada label bukan *main content* dan 0.59 pada label *main content* maka item tersebut diklasifikasikan sebagai *main content*.

Probability Cut-Off Point adalah nilai probabilitas yang digunakan jika ingin mendapatkan nilai *sensitivity* dan *specificity* terbaik untuk masing-masing label. Dengan menggunakan 4 *Feature Set* terlihat bahwa nilai *Probability Cut-Off point* yang didapat adalah sebesar 0.998 untuk label bukan *main content* dan 0.0001 untuk label *main content* pada halaman web. Sedangkan untuk halaman web yang

Tabel 4.15: Hasil Correlation Matrix Untuk 11 Fitur

	words	sentence	link	TD	max_cw	mean_cw	tf	tbf	pf	lf	max_oc
words	1.00	0.89	0.42	0.61	0.47	0.32	0.45	0.63	0.61	0.45	0.90
sentence	0.89	1.00	0.11	0.46	0.36	0.22	0.44	0.79	0.59	0.12	0.80
link	0.42	0.11	1.00	0.44	0.14	0.10	0.18	0.01	0.05	0.93	0.40
TD	0.61	0.46	0.44	1.00	0.36	0.39	0.09	0.42	0.11	0.49	0.66
max_cw	0.47	0.36	0.14	0.36	1.00	0.83	0.29	0.05	0.26	0.14	0.41
mean_cw	0.32	0.22	0.10	0.39	0.83	1.00	0.12	0.01	0.09	0.12	0.30
tf	0.45	0.44	0.18	0.09	0.29	0.12	1.00	0.12	0.57	0.13	0.39
tbf	0.63	0.79	0.01	0.42	0.05	0.01	0.12	1.00	0.19	0.01	0.66
pf	0.61	0.59	0.05	0.11	0.26	0.09	0.57	0.19	1.00	0.07	0.45
lf	0.45	0.12	0.93	0.49	0.14	0.12	0.13	0.01	0.07	1.00	0.42
max_oc	0.90	0.80	0.40	0.66	0.41	0.30	0.39	0.66	0.45	0.42	1.00

Tabel 4.16: Nilai P-Value dan T-Value Untuk 11 Fitur

Fitur	T-Value	P-Value
words	12.079	2.22e-308
sentence	7.6382	2.22e-308
link	16.623	5.9e-148
TD	19.425	2.22e-308
max_cw	27.121	2.22e-308
mean_cw	38.27	2.22e-308
tf	10.427	2.22e-308
tbf	2.8471	5.7e-196
pf	4.5487	2.22e-308
lf	15.459	3.35e-134
max_oc	15.4592	2.22e-308

telah diperbaiki Tidy didapat *Probability Cut-Off Point* sebesar 0.9965 untuk label bukan main content dan 0.0003 untuk label *main content*.

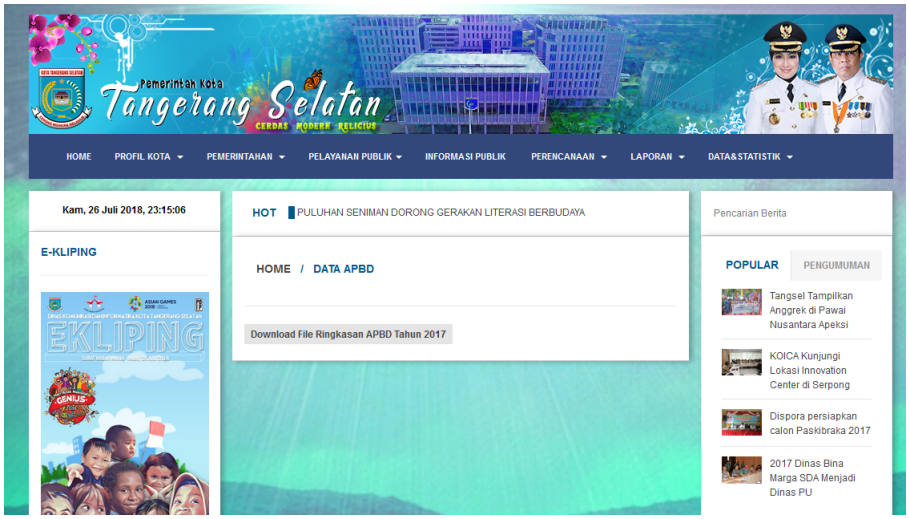
Sebagai gambaran ketika *probability threshold* diset menjadi 1 atau dengan kata lain sebuah blok atau segmen dikatakan sebagai *main content* jika probabilitas yang yang dihasilkan oleh model klasifikasi untuk blok tersebut adalah 100% *main content*, dengan kata lain model klasifikasi tersebut hanya mengklasifikasikan model tersebut sebagai *main content* jika dan hanya jika model tersebut yakin 100% bahwa blok tersebut merupakan *main content*. Model yang sempurna akan mendapat tingkat Sensitivity 1 atau 100% dan tingkat Specificity 1 atau 100% ketika tingkat *probability threshold* diset menjadi 100%.

Dengan melihat grafik *specificity* dan *sensitivity* yang ada pada gambar 4.9 dan gambar 4.10 dan *Probability Cut Off point* yang terbentuk pada kedua gambar tersebut, maka dapat diambil kesimpulan bahwa model yang terbentuk dapat mendapatkan tingkat *sensitivity* dan *specificity* yang baik untuk label bukan *main content* dengan nilai *probability cut-off point* yang cukup tinggi (0.998 dan 0.9965) dimana dapat ditarik kesimpulan bahwa model yang telah dibangin dapat mengenali blok atau segmen yang bukan *main content* dengan sangat baik.

Sebaliknya terlihat bahwa model tidak dapat mengindentifikasi label *main content* dengan baik. Hal ini terlihat dari rendahnya nilai *probability cut-off point* yang didapat untuk label *main content* yaitu sebesar 0.0001 dan 0.0003. Selain jika dili-

hat pada gambar 4.9 dan gambar 4.10, terlihat bahwa pada label bukan *main content* memiliki grafik *Specificity* yang cukup buruk dimana pada label *main content* memiliki grafik *Sensitivity* yang buruk. Hal tersebut menandakan bahwa banyak label *main content* yang *misclassified* atau diklasifikasikanan sebagai label bukan *main content*. Hal ini akan bermasalah karena untuk penelitian ini akan lebih berfokus untuk melakukan pengambilan *main content*.

Selain itu perlu dilihat mengenai nilai *probability cut-off point* untuk label *main content* yang sangat rendah yaitu 0.0001 dan 0.0003. Hal ini menandakan bahwa terdapat blok atau segmen *main content* yang memiliki fitur yang sangat menyerupai blok yang bukan *main content*. Berberapa contoh yang dapat menyebabkan sebuah blok memiliki fitur yang sangat mirip dengan blok yang bukan *main content* diantaranya adalah blok tersebut memiliki *main content* terenkapsulasi oleh teknologi iframe seperti pada contoh kode 4.5.

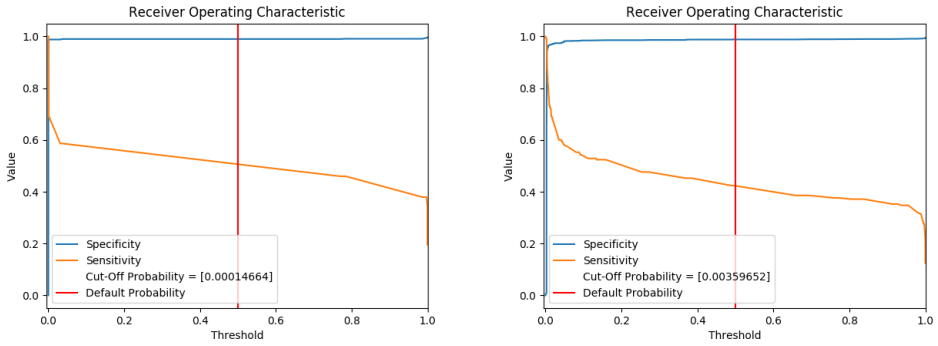


Gambar 4.8: Main Content Yang Sulit Di-identifikasi

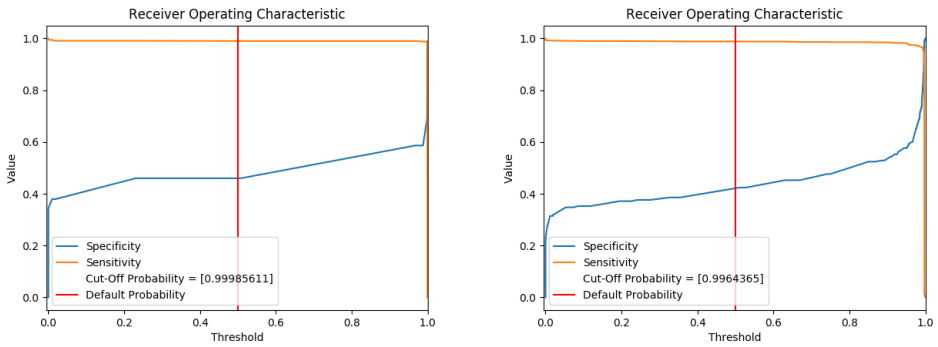
Contoh lain mengenai blok yang *main content* yang memiliki fitur yang menyerupai blok yang bukan *main content* adalah blok *main content* tersebut hanya berisi sebuah link tanpa adanya keterangan atau penjelasan seperti yang terlihat pada gambar 4.8. Pada gambar 4.8, terlihat halaman web hanya berisi sebuah kata dan link untuk men-*download* sebuah perda, dimana fitur yang didapat dari blok seperti ini akan sangat mirip dengan blok yang merupakan bukan *main content*. Hal yang

dapat dilakukan oleh pemerintah daerah adalah dengan memusatkan konten-konten seperti link *download* mengenai perda pada sebuah halaman web yang berisi daftar perda yang dapat di-*download*.

Selanjutnya ketika menggunakan 11 fitur set seperti yang terlihat pada gambar 4.11 dan gambar 4.12. Terlihat terdapat peningkatan dari segi grafik *Specificity* pada label bukan *main content* dan grafik *Sensitivity* pada label *main content*. Hal ini menandakan bahwa permasalahan *misclassified* pada label *main content* semakin berkurang.

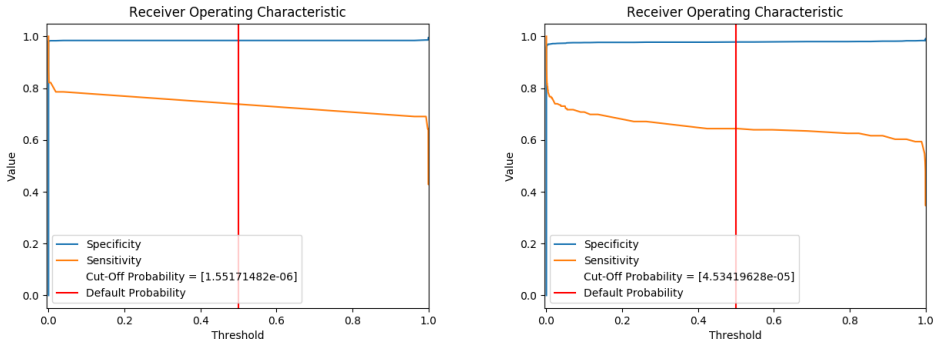


Gambar 4.9: Grafik ROC *Sensitivity* dan *Specificity* untuk 4 *Feature Set* untuk Label *Main Content* untuk halaman web (kiri) dan halaman web yang telah diperbaiki Tidy (Kanan)

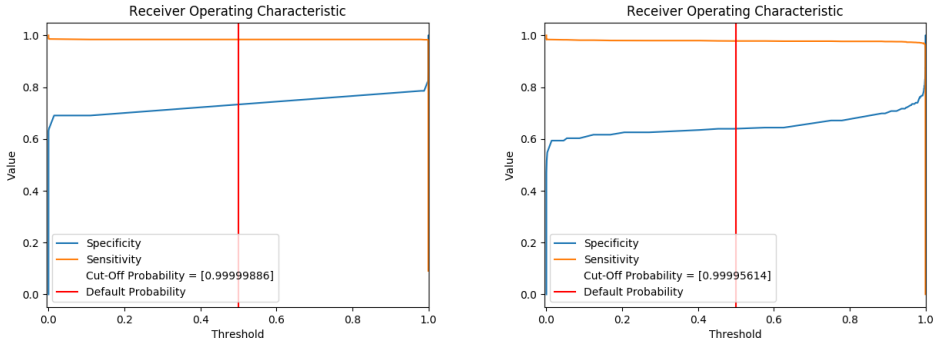


Gambar 4.10: Grafik ROC *Sensitivity* dan *Specificity* untuk 4 *Feature Set* untuk Label bukan *Main Content* untuk halaman web (kiri) dan halaman web yang telah diperbaiki Tidy (Kanan)

Hasil *confusion matrix* ketika menggunakan 11 fitur dapat dilihat pada tabel 4.22 dan hasil evaluasi dapat dilihat pada tabel 4.18 untuk halaman web dan halaman web yang telah diperbaiki tidy. Terlihat bahwa terdapat peningkatan *precision*



Gambar 4.11: Grafik ROC *Sensitivity* dan *Specificity* untuk 11 *Feature Set* untuk Label *Main Content* untuk halaman web (kiri) dan halaman web yang telah diperbaiki Tidy (Kanan)



Gambar 4.12: Grafik ROC *Sensitivity* dan *Specificity* untuk 11 *Feature Set* untuk Label bukan *Main Content* untuk halaman web (kiri) dan halaman web yang telah diperbaiki Tidy (Kanan)

dan *recall* pada label *main content* dibandingkan hasil yang didapat pada tabel 4.4 untuk halaman web dan 4.5 untuk halaman web yang telah diperbaiki Tidy.

Tabel 4.17: Hasil *Confusion Matrix* untuk Model dengan menggunakan 11 Fitur

		Halaman Web		Halaman Web yang telah diperbaiki Tidy	
		Predicted		Predicted	
		Yes	No	Yes	No
Actual	Yes	62	22	140	79
	No	17	1053	50	2528

Tabel 4.18: Hasil Evaluasi untuk Model dengan menggunakan 11 Fitur

	Halaman Web			Halaman Web yang telah diperbaiki Tidy		
	precision	recall	f1-score	precision	recall	f1-score
Main Content	0.78	0.74	0.76	0.74	0.66	0.7
Bukan Main Content	0.98	0.98	0.98	0.97	0.98	0.98
avg/total	0.97	0.97	0.97	0.95	0.95	0.95
Accuracy	0.966204506			0.954951735		

4.1.3.3 Balancing Dataset

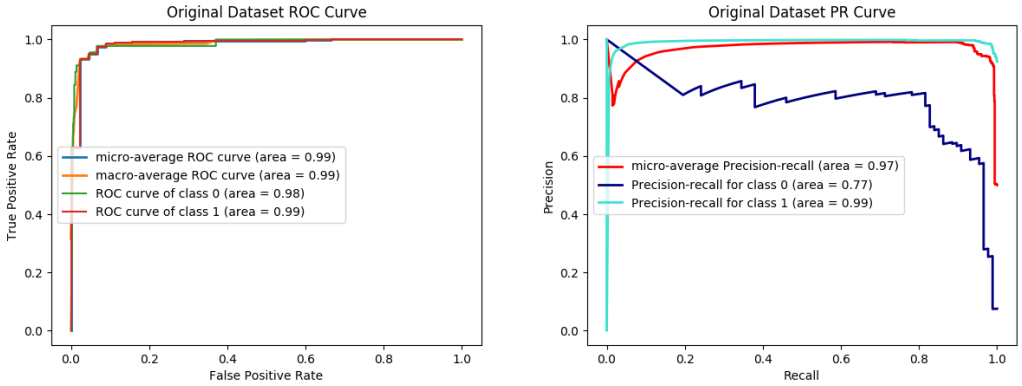
Dengan melihat nilai *recall* dan *precision* yang dihasilkan pada label main content pada tabel 4.18, maka perlu dilakukan analisis lebih lanjut mengenai model yang telah dibuat untuk meningkatkan nilai *precision* dan *recall* yang didapat pada label *main content*. Lebih lanjut dengan melihat dataset yang telah terbentuk terlihat bahwa terdapat ketidakseimbangan pada label *main content* dengan label bukan *main content* dimana mencapai perbandingan 1:10 untuk halaman web dan 1:12 untuk halaman web yang telah diperbaiki tidy, seperti yang terlihat pada tabel 4.19. Hal ini menunjukkan bahwa dataset yang telah terbentuk pada tahap pengambilan *main content* menggunakan *template-based* menghasilkan data yang merupakan *imbalanced* dataset.

Tabel 4.19: Perbandingan data pada setiap label

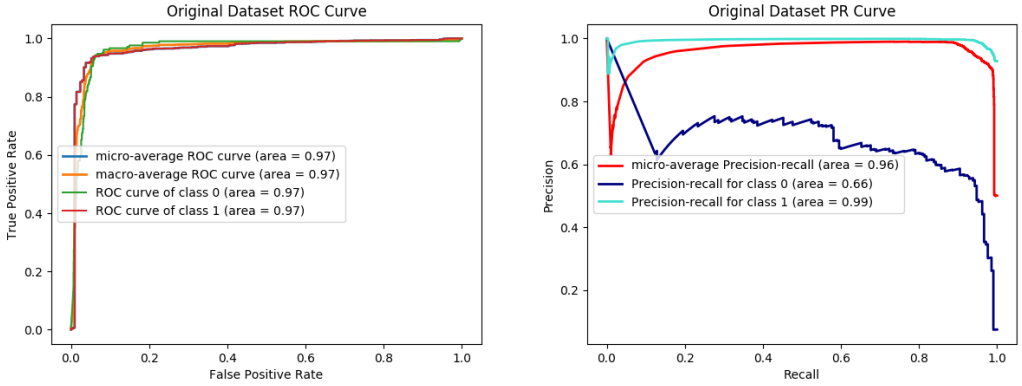
	Halaman Web	Halaman Web yang telah diperbaiki Tidy
Main Content	541	1009
Bukan Main Content	5044	12309
Total	5585	13318

Berberapa penelitian terdahulu menjelaskan bahwa salah satu metrik pengukuran yang dapat digunakan untuk menilai performa dari model dengan imbalanced dataset adalah dengan melihat grafik ROC (Receiver Operating Characteristics) *Curve* dan PR (Precision-Recall) *Curve* [Hoens and Chawla, 2013, Saito and Rehmsmeier, 2015, Jeni et al., 2013]. Pada gambar 4.13 dan gambar 4.14 merupak-

an grafik dari ROC *Curve* dan PR *Curve* dengan menggunakan 4 *Feature Set* dan menggunakan pembagian 7:3 dari total dataset yang dipergunakan untuk *training* untuk melakukan *training* dan validasi model pada halaman web dan halaman web yang telah diperbaiki Tidy. Sedangkan pada gambar 4.15 dan gambar 4.16 merupakan grafik dari ROC *Curve* dan PR *Curve* dengan menggunakan 11 *Feature Set* pada halaman web dan halaman web yang telah diperbaiki Tidy.

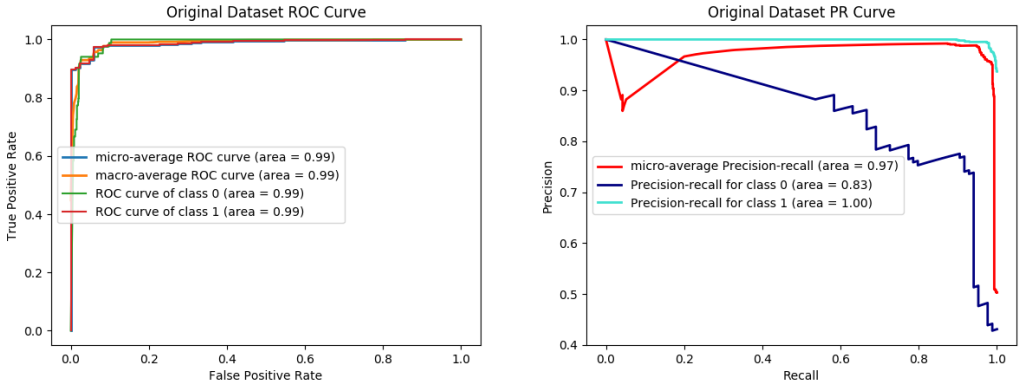


Gambar 4.13: PR Curve dan ROC Curve untuk Halaman Web dengan 4 *Feature Set*

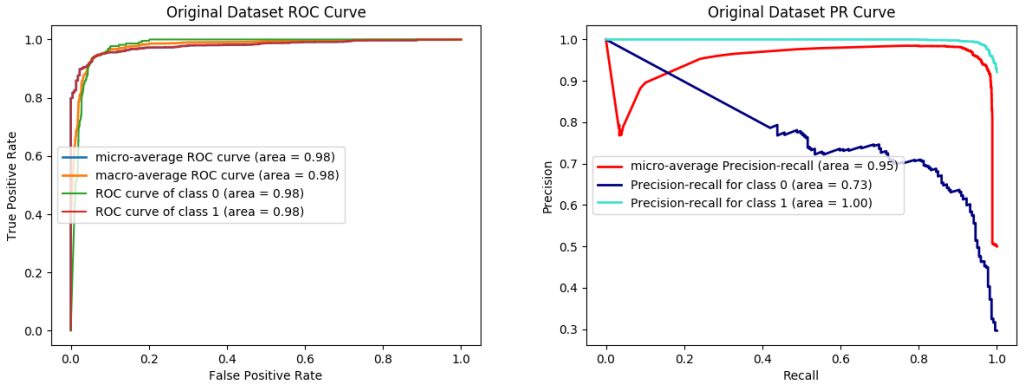


Gambar 4.14: PR Curve dan ROC Curve untuk Halaman Web yang telah diperbaiki tidy dengan 4 *Feature Set*

Pada gambar 4.13 dan gambar 4.14 menunjukkan bahwa ROC *curve* memiliki hasil yang baik dengan nilai AUC (Area Under Curve) ROC rata-rata diatas 0.97 untuk halaman web dan halaman web yang telah diperbaiki Tidy, sedangkan PR *curve* yang dihasilkan memiliki hasil yang kurang baik terutama untuk label class



Gambar 4.15: PR Curve dan ROC Curve untuk Halaman Web dengan 11 *Feature Set*



Gambar 4.16: PR Curve dan ROC Curve untuk Halaman Web yang telah diperbaiki tidy dengan 11 *Feature Set*

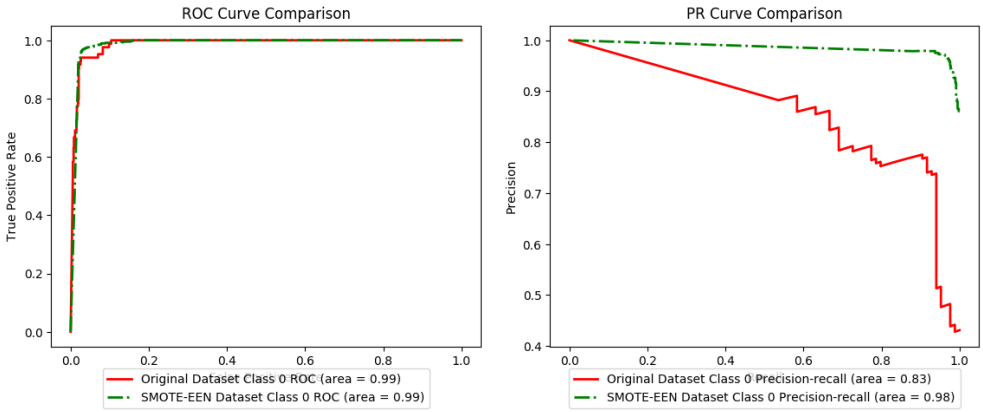
0 atau label *main content* yang memiliki nilai AUC PR sebesar 0.77 untuk halaman web dan 0.66 untuk halaman web yang telah diperbaiki Tidy.

Sedangkan pada gambar 4.15 dan gambar 4.16 menunjukkan terjadinya peningkatan AUC PR pada label *main content*, dimana AUC PR pada halaman web memiliki nilai sebesar 0.83 dari yang sebelumnya bernilai 0.77 dan AUC PR pada halaman web yang telah diperbaiki tidy memiliki nilai sebesar 0.73 dari yang sebelumnya bernilai 0.66

Terlihat bahwa dengan memodifikasi atau menambah feature set dapat meningkatkan akurasi yang dihasilkan oleh model klasifikasi yang dibuat akan tetapi permasalahan *imabalanced dataset* masih mempengaruhi akurasi dari model klasifikasi yang dibangun dengan signifikan. Untuk mengatasi permasalahan mengenai

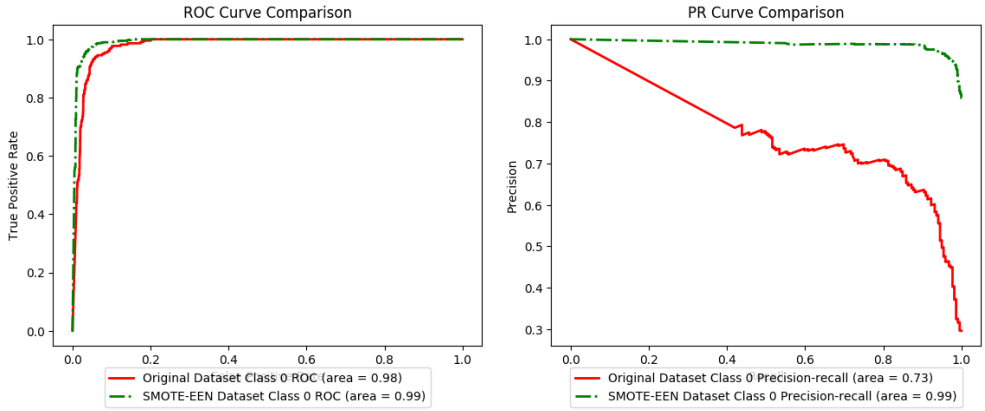
imbalanced dataset tersebut maka pada penelitian ini juga dilakukan *balancing* terhadap *dataset* yang dihasilkan dari tahap pengambilan *main content* dengan pendekatan *template-based* untuk membentuk model yang lebih baik dan akurat terutama model yang memiliki nilai *recall* dan *precision* yang baik pada label *main content*. Batista mengatakan bahwa dengan melakukan *balancing* terhadap *dataset* yang digunakan merupakan salah satu solusi positif dalam menangani permasalahan *imbalanced* dataset [Batista et al., 2004].

Pada penelitian ini, dilakukan metode *dataset balancing* menggunakan metode SMOTE-EEN. Untuk pembentukan model dengan menggunakan *balancing dataset* ini digunakan pembagian 7:3 dari total *dataset* yang dipergunakan untuk *training* dari tahap pengambilan *main content* dengan pendekatan *template-based*, dimana 70% *dataset* akan digunakan untuk membentuk *training* dan 30% *dataset* akan digunakan untuk validasi. Selanjutnya akan dilakukan komparasi hasil yang didapat ketika menggunakan *dataset balancing* SMOTE-EEN dibandingkan tanpa menggunakan *dataset balancing*.

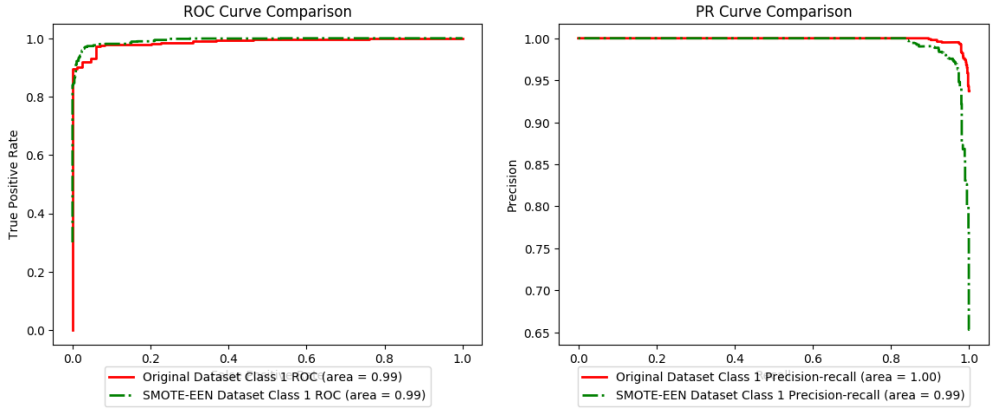


Gambar 4.17: Komparasi Metode *Dataset Balancing* untuk label *main content* pada Halaman Web dengan 11 *Feature Set*

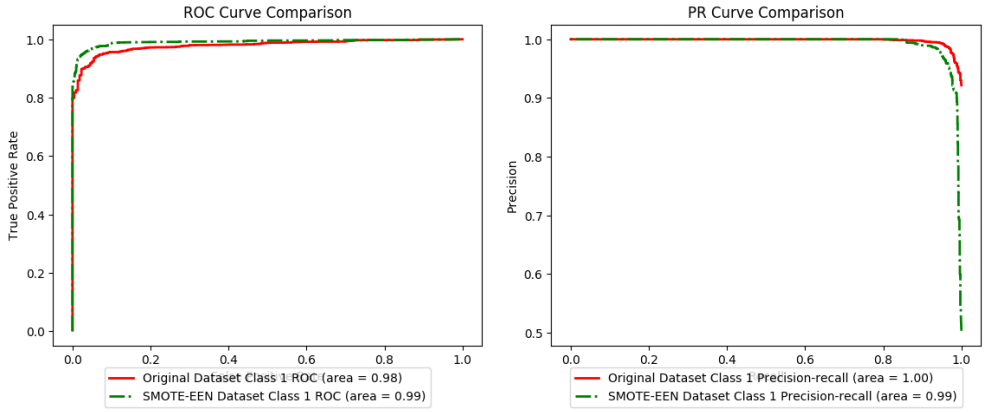
Pada Gambar 4.23 dan gambar 4.24 terlihat bahwa sekali lagi terjadi peningkatan dari segi grafik *Specificity* pada label bukan *main content* dan grafik *Sensitivity* pada label *main content*. Hal ini dapat menunjukkan bahwa permasalahan *misclassified* pada label *main content* semakin berkurang dibandingkan hasil yang didapat dari gambar 4.11 dan gambar 4.12.



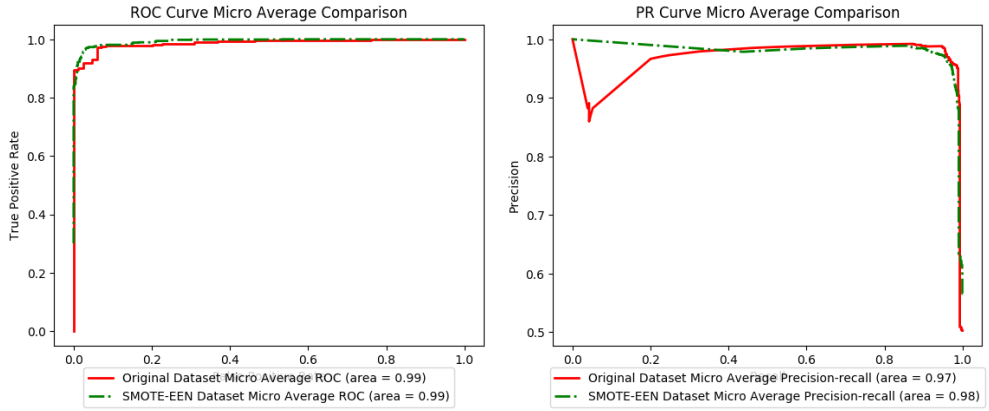
Gambar 4.18: Komparasi Metode *Dataset Balancing* untuk label *main content* pada Halaman Web yang telah diperbaiki Tidy dengan 11 *Feature Set*



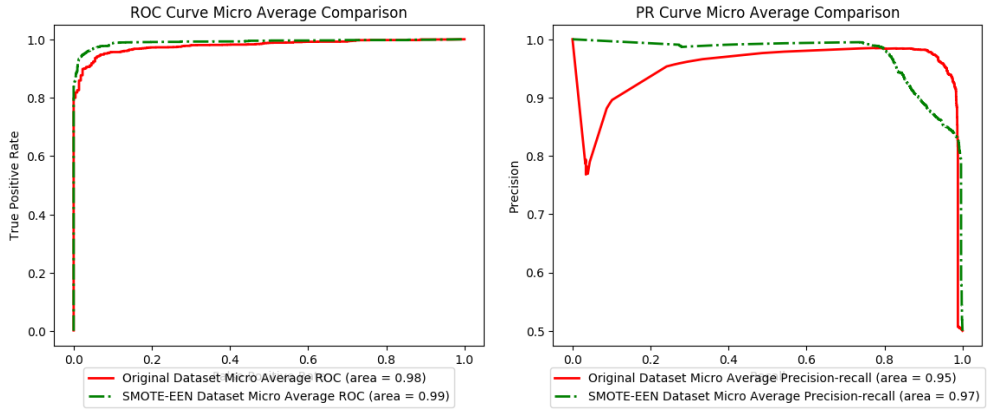
Gambar 4.19: Komparasi Metode *Dataset Balancing* untuk label bukan *main content* pada Halaman Web dengan 11 *Feature Set*



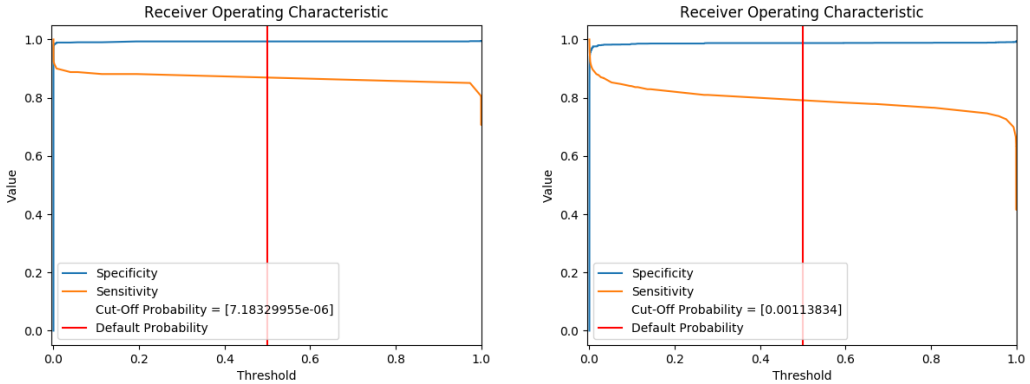
Gambar 4.20: Komparasi Metode *Dataset Balancing* untuk label bukan main content pada Halaman Web yang telah diperbaiki Tidy dengan 11 *Feature Set*



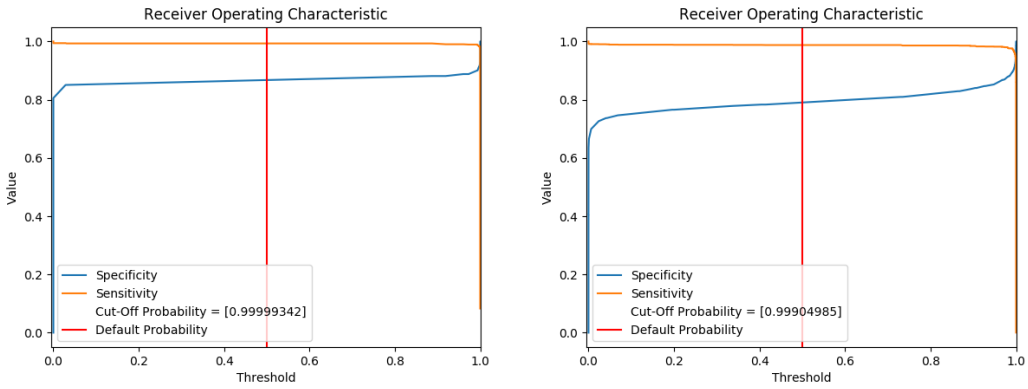
Gambar 4.21: Komparasi Metode *Dataset Balancing* untuk *Micro Average* pada Halaman Web dengan 11 *Feature Set*



Gambar 4.22: Komparasi Metode *Dataset Balancing* untuk *Micro Average* pada Halaman Web yang telah diperbaiki Tidy dengan 11 *Feature Set*



Gambar 4.23: Grafik ROC *Sensitivity* dan *Specificity* untuk 11 *Feature Set* untuk Label *Main Content* pada setiap *Probability Threshold* untuk halaman web (kiri) dan halaman web yang telah diperbaiki Tidy (Kanan)



Gambar 4.24: Grafik ROC *Sensitivity* dan *Specificity* untuk 11 *Feature Set* untuk Label *Main Content* pada setiap *Probability Threshold* untuk halaman web (kiri) dan halaman web yang telah diperbaiki Tidy (Kanan)

Pada gambar 4.17 hingga gambar 4.22 merupakan komparasi dari ROC *Curve* dan PR *Curve* yang dihasilkan dari metode *dataset balancing* yang digunakan untuk *dataset* dengan menggunakan sebelas fitur pada halaman web dan halaman web yang telah dipebariki dengan tidy. Dari hasil komparasi balancing tersebut terlihat bahwa metode SMOTE-EEN memberikan hasil yang lebih untuk masing-masing label dan hasil *micro-average* dibandingkan tanpa menggunakan *dataset balancing*.

Tabel 4.20: Perbandingan *Confusion Matrix* untuk Model dengan SMOTE-EEN

		Halaman web		Halaman Web yang telah diperbaiki Tidy	
		Predicted		Predicted	
		Original			
		Yes	No	Yes	No
Actual	Yes	62	22	140	79
	No	17	1053	50	2528
		SMOTE-EEN			
Actual	Yes	989	51	2008	511
	No	28	1020	24	2531

Hasil dari *confusion matrix* dapat dilihat pada tabel 4.20, Selain itu nilai *recall* dan *precision* pada *classification report* yang diperoleh ketika menggunakan *dataset balancing* SMOTE-EEN memiliki nilai yang labih baik terutama untuk ni-

Tabel 4.21: Perbandingan Evaluasi Model dengan SMOTE-EEN

	Tanpa SMOTE-EEN			SMOTE-EEN		
	Main Content	Bukan Main Content	Avg / Total	Main Content	Bukan Main Content	Avg / Total
	Halaman Web					
Precision	0.78	0.98	0.97	0.97	0.95	0.96
Recall	0.74	0.98	0.97	0.95	0.97	0.96
F1-Score	0.76	0.98	0.97	0.96	0.96	0.96
Accuracy	0.966204506			0.962164751		
	Halaman Web yang telah diperbaiki Tidy					
Precision	0.74	0.97	0.95	0.99	0.83	0.91
Recall	0.64	0.98	0.95	0.8	0.99	0.89
F1-Score	0.68	0.98	0.95	0.88	0.9	0.89
Accuracy	0.953879156			0.894560505		

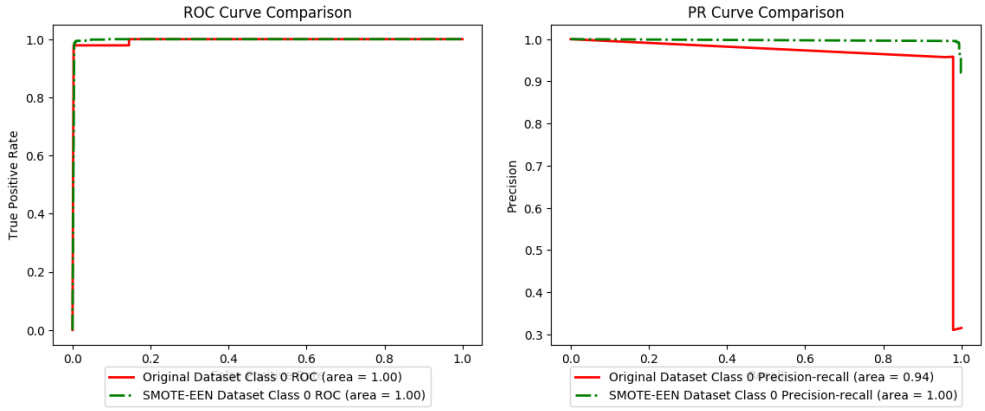
lai *precision* dan *recall* pada label *main content*, seperti yang terlihat pada tabel 4.21, apabila dibandingkan adengan hasil yang didapat tanpa menggunakan *dataset balancing* SMOTE-EEN pada tabel 4.18. Peningkatan hasil yang didapatkan oleh SMOTE-EEN bisa disebabkan karena SMOTE-EEN melakukan *over sampling* pada label *main content* dimana pada umumnya fitur yang dimiliki oleh label *main content* akan berbeda dibandingkan label yang bukan *main content* dan ketika *data over sampling* yang terbentuk menyerupai label *main content* maka data tersebut akan difilter oleh ENN.

4.1.3.4 Filterisasi Dataset

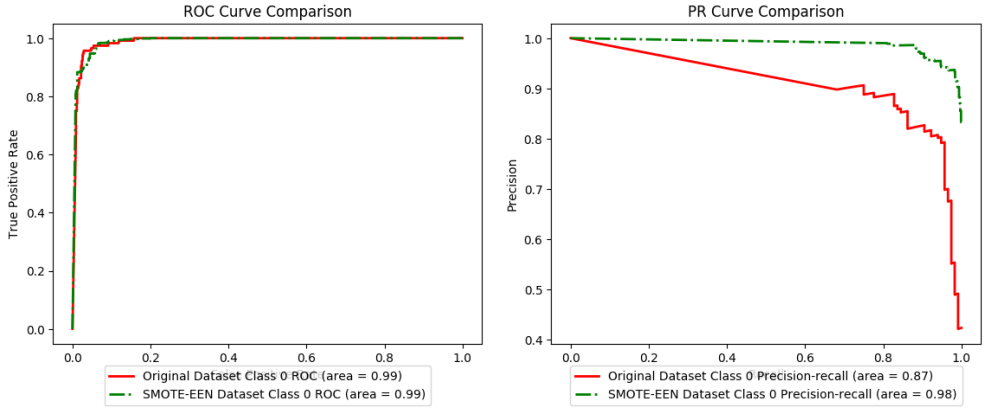
Pada pembahasan mengenai pengambilan *main content* dengan menggunakan *template based* telah dibahas mengenai permasalahan-permasalahan yang mengakibatkan pengambilan kandidat *main content* berjalan dengan tidak sempurna.

Dengan mengganggap bahwa data yang terbentuk ketika permasalahan - permasalahan tersebut terjadi sebagai *noisy data*, maka pada penelitian ini juga coba dilakukan pembentukan model klasifikasi dengan hanya menggunakan dataset yang tidak memiliki permasalahan yang diutarakan pada tahap pengambilan *main content* dengan menggunakan *template-based*. Hal ini dilakukan untuk melihat apakah da-

ta yang bermasalah tersebut mempunyai dampak yang cukup signifikan terhadap model yang dihasilkan.



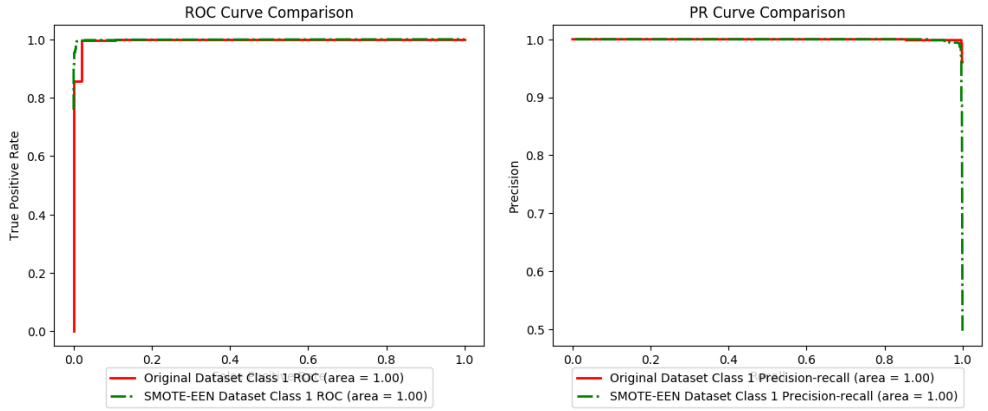
Gambar 4.25: Komparasi Hasil Filterisasi pada label *main content* pada Halaman Web dengan 11 *Feature Set*



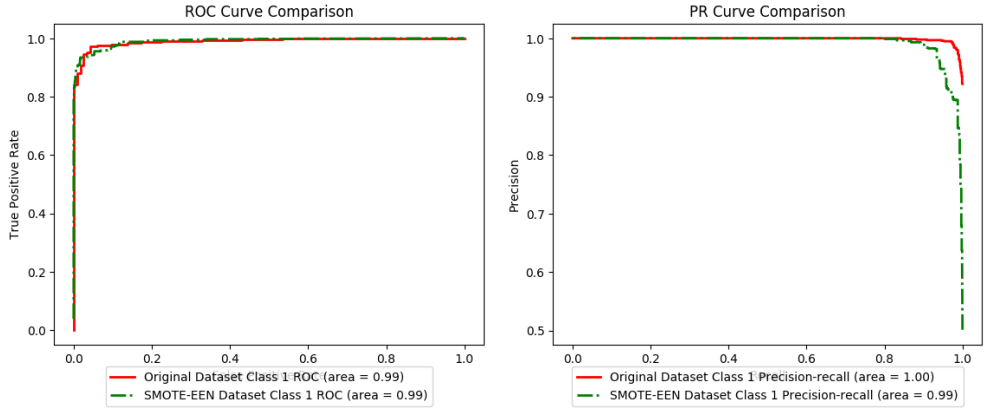
Gambar 4.26: Komparasi Hasil Filterisasi pada label *main content* pada Halaman Web yang telah diperbaiki Tidy dengan 11 *Feature Set*

Pada Gambar 4.31 dan gambar 4.32 terlihat bahwa sekali lagi terjadi peningkatan dari segi grafik *Specificity* pada label bukan *main content* dan grafik *Sensitivity* pada label *main content*. Hal ini dapat menunjukkan bahwa permasalahan *misclassified* pada label *main content* semakin berkurang dibandingkan hasil yang didapat dari gambar 4.23 dan gambar 4.24.

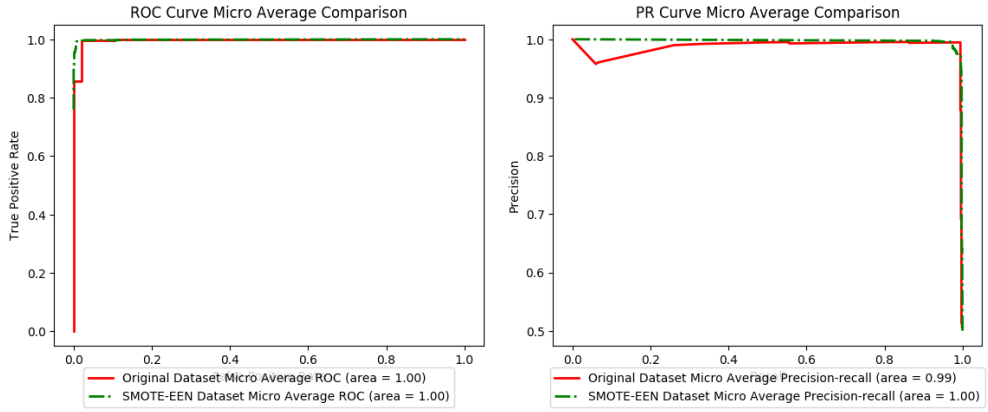
Pada gambar 4.25 hingga gambar 4.30 merupakan komparasi dari ROC *Curve* dan PR *Curve* yang dihasilkan dari hasil filterisasi dan menggunakan *dataset balancing* SMOTE-EEN untuk *dataset* yang dihasilkan dengan 11 fitur. Selain itu



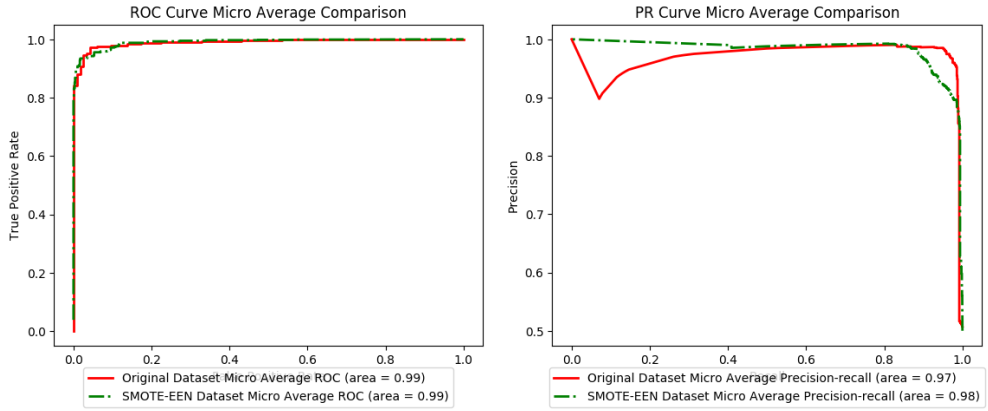
Gambar 4.27: Komparasi Hasil Filterisasi pada label bukan *main content* pada Halaman Web dengan 11 *Feature Set*



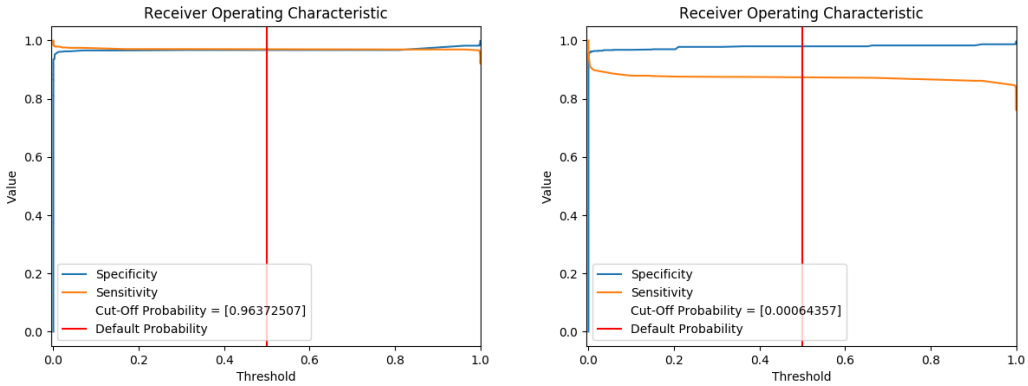
Gambar 4.28: Komparasi Hasil Filterisasi pada label bukan *main content* pada Halaman Web yang telah diperbaiki Tidy dengan 11 *Feature Set*



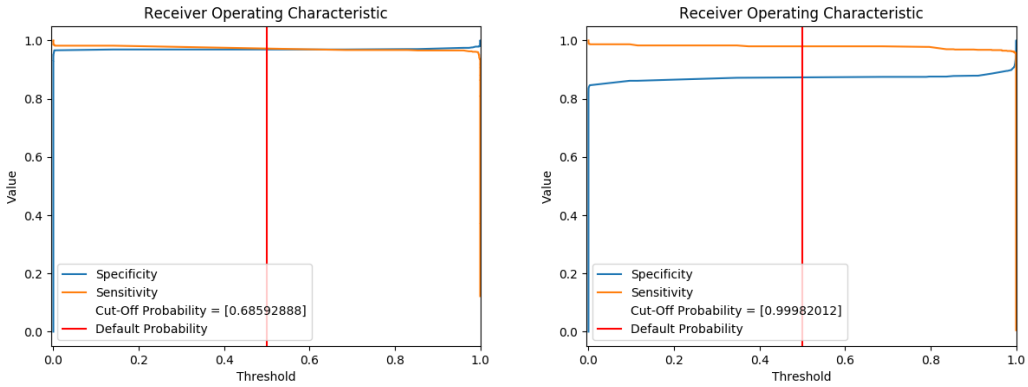
Gambar 4.29: Komparasi Hasil Filterisasi pada *Micro Average* pada Halaman Web dengan 11 *Feature Set*



Gambar 4.30: Komparasi Hasil Filterisasi pada *Micro Average* pada Halaman Web yang telah diperbaiki Tidy dengan 11 *Feature Set*



Gambar 4.31: Grafik ROC *Sensitivity* dan *Specificity* untuk 11 *Feature Set* untuk Label *Main Content* pada setiap *Probability Threshold* untuk halaman web (kiri) dan halaman web yang telah diperbaiki Tidy (Kanan)



Gambar 4.32: Grafik ROC Sensitivity dan Specificity untuk 11 *Feature Set* untuk Label *Main Content* pada setiap *Probability Threshold* untuk halaman web (kiri) dan halaman web yang telah diperbaiki Tidy (Kanan)

hasil *confusion matrix* , seperti yang terlihat pada tabel 4.22, dan hasil evaluasi dari model yang terbentuk , seperti yang terlihat pada tabel 4.23, memperlihatkan bahwa hasil yang didapat sangat baik terutama untuk nilai *precision* dan *recall* pada label *main content* memberikan nilai yang lebih baik dibandingkan hasil yang didapatkan model yang dibangun tanpa melakukan filterisasi seperti pada tabel 4.21.

Tabel 4.22: Perbandingan *Confusion Matrix* Model dengan Filterisasi

		Halaman web		Halaman Web yang telah diperbaiki Tidy	
		Predicted		Predicted	
		Original			
		Yes	No	Yes	No
Actual	Yes	46	1	97	19
	No	3	701	16	988
		SMOTE-EEN			
Actual	Yes	692	4	885	97
	No	23	665	28	959

Tabel 4.23: Perbandingan Evaluasi Model dengan Filterisasi

	Filterisasi Tanpa SMOTE-EEN			Filterisasi dan SMOTE-EEN		
	Main Content	Bukan Main Content	Avg / Total	Main Content	Bukan Main Content	Avg / Total
	Halaman Web					
Precision	0.86	0.99	0.98	0.67	1	0.97
Recall	0.9	0.99	0.98	0.98	0.96	0.96
F1-Score	0.88	0.99	0.98	0.79	0.98	0.96
Accuracy	0.966204506			0.962164751		
	Halaman Web yang telah diperbaiki Tidy					
Precision	0.86	0.98	0.97	0.97	0.91	0.94
Recall	0.84	0.98	0.97	0.9	0.97	0.94
F1-Score	0.85	0.98	0.97	0.93	0.94	0.94
Accuracy	0.953879156			0.894560505		

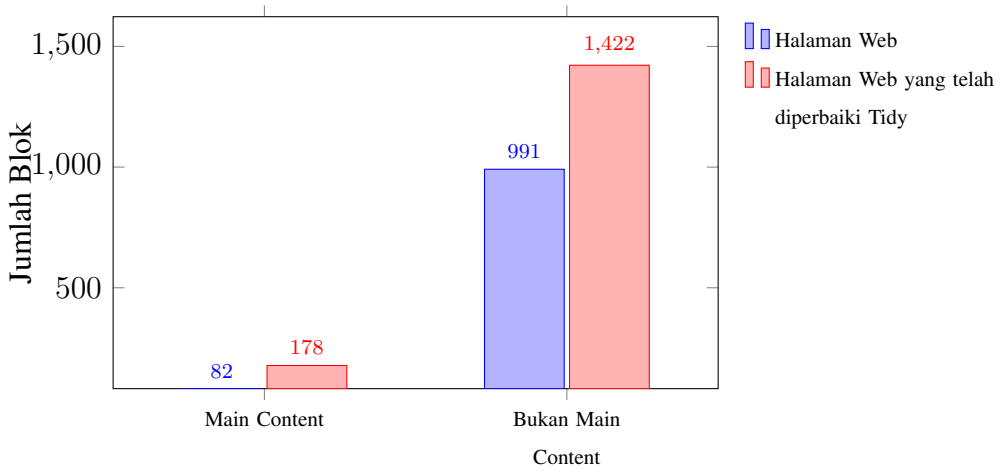
4.2 Pengujian dan Evaluasi

Pada bagian berikut ini akan dijelaskan mengenai pengujian yang dilakukan dan pembahasan mengenai hasil yang didapat dari pengujian tersebut.

4.2.1 Pengujian dengan hanya menggunakan pembagian blok hasil pendekatan *template-based* untuk melakukan pengambilan *main content*

Pengujian Pertama yaitu dengan hanya menggunakan blok yang dihasilkan dari pendekatan *template-based* untuk melakukan pengambilan *main content*. Pengujian ini dilakukan dengan menggunakan 30% dari *dataset* yang telah dibagi pada tahap pembentukan model klasifikasi *machine learning*. Hasil pada pengujian ini memperlihatkan bahwa *main content* dapat diambil secara keseluruhan untuk setiap halaman web. Walaupun demikian blok atau segmen yang merupakan bukan *main content* memiliki jumlah yang signifikan lebih tinggi daripada blok atau segmen yang bukan *main content*. seperti yang terlihat pada gambar 4.33.

Hal ini bisa diakibatkan oleh permasalahan mengenai perbedaan kecil pada *layout* pada berberapa halaman web yang ada pada satu situs web resmi pemerintah daerah yang telah dibahas pada tahap pengambilan *main content* dengan menggunakan pendekatan *template-based*. Pada penelitian ini *threshold* atau batas minimal kesamaan *node* yang digunakan pada saat proses pendeteksian *template* adalah sebanyak jumlah halaman web yang diproses.



Gambar 4.33: Perbandingan Blok yang Didapat Saat Menggunakan Pendekatan Template-Based

Sebagai contoh jika sebuah situs web resmi pemerintah daerah memiliki 9 halaman web yang telah lolos validasi yang kemudian akan diproses pada tahap pengambilan *main content* menggunakan pendekatan *template-based* sehingga halaman web yang diproses adalah sebanyak sembilan halaman web. Apabila dari sembilan halaman web tersebut terdapat satu halaman web yang memiliki *layout* yang sedikit berbeda maka seperti yang terlihat pada gambar 4.6 maka blok tersebut akan dianggap sebagai sebuah blok yang unik.

Permasalahan tersebut dapat menimbulkan 2 hasil yang berbeda sesuai dengan struktur HTML dari halaman web tersebut. Hasil pertama yaitu 8 blok tersebut dianggap menjadi blok unik yang terpisah pada 9 halaman web tersebut dan menambah jumlah blok atau segmen yang bukan *main content*. Sedangkan hasil kedua yaitu *parent* dari blok tersebut dianggap menjadi sebuah blok unik pada 9 halaman web tersebut yang mengakibatkan *feature set* dari blok *main content* berubah menyerupai blok yang bukan *main content*. Dengan menyesuaikan nilai dari *threshold* atau batas minimal kesamaan *node* dalam proses pendeteksian *template* maka jumlah dari blok atau segmen yang bukan *main content* dapat dikurangi atau dioptimalkan.

Meskipun demikian untuk mendapatkan nilai *threshold* atau batas minimal kesamaan *node* perlu dilakukan dengan tepat, misalnya apabila batas nilai *threshold* atau batas minimal kesamaan *node* diset menjadi 2, maka *main content* pada halaman web yang memiliki kemiripan struktur dalam penyajian main content akan terambil secara tidak sempurna misalnya profil pejabat pemerintahan A pada halaman web A dan profil pejabat pemerintahan B pada halaman web B akan memiliki kesamaan pola dalam penyajian *main content* seperti adanya kolom nama atau nomor induk pegawai dimana terdapat kemungkinan jika kolom tersebut dianggap sebagai bukan *main content*. Contoh lain adalah jika sebuah tabel pada 2 halaman web memiliki *heading* yang sama maka *heading* tersebut kemungkinan akan dianggap sebagai bukan *main content*. Sehingga untuk penelitian kedepannya dapat dilakukan pengoptimalan nilai dari *threshold* atau batas minimal kesamaan *node*.

4.2.2 Pengujian dengan menggunakan blok hasil pendekatan *template-based* dan klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi untuk melakukan pengambilan *main content*

Pengujian Kedua yaitu pengujian mengenai pengambilan *main content* dengan blok hasil pendekatan *template-based* dan klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi. Pengujian ini menggunakan model yang dibentuk dengan menggunakan 11 fitur, filterisasi dan *dataset balancing* SMOTE-EEN, yang kemudian dilakukan komparasi dengan Model awal yang terbentuk dengan menggunakan 4 Fitur tanpa melakukan filterisasi dan tanpa melakukan *dataset balancing* SMOTE-EEN.

Tabel 4.24: Hasil Komparasi *Confusion Matrix* pada Pengujian Kedua

		Halaman web		Halaman yang telah diperbaiki Tidy	
		Predicted		Predicted	
		Model Awal			
		Yes	No	Yes	No
Actual	Yes	57	72	49	80
	No	21	1499	21	1499
		Model yang telah di Perbaiki			
Actual	Yes	137	0	131	6
	No	138	1374	78	1434

Dengan melihat hasil yang muncul *confusion matrix* pada tabel 4.24 dan hasil evaluasi pada tabel 4.25, terlihat bahwa dengan melakukan penambahan fitur, filterisasi dataset dan balancing dataset SMOTE-EEN dapat memberikan hasil yang baik ketika mengidentifikasi label yang merupakan *main content*. Peningkatan hasil yang dicapai dengan melakukan penambahan fitur, filterisasi dataset dan balancing dataset SMOTE-EEN bila dibandingkan dengan model awal yang terbentuk dengan tanpa melakukan penambahan fitur, filterisasi dataset dan balancing dataset SMOTE-EEN bisa diattributkan oleh berberapa faktor, diantaranya :

1. Dengan melakukan penambahan fitur menjadi 11 fitur dari yang sebelumnya 4 fitur, dapat memberikan gambaran atau pola yang lebih mendalam pada

Tabel 4.25: Hasil Komparasi Evaluasi pada Pengujian Kedua

	Halaman Web			Halaman Web yang telah diperbaiki Tidy		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
	Model Awal					
Main Content	0.73	0.44	0.55	0.70	0.38	0.49
Bukan Main Content	0.95	0.99	0.97	0.95	0.99	0.97
Avg Total	0.94	0.94	0.94	0.93	0.94	0.93
Accuracy	9436021			0.94529		
	Model Yang telah di Perbaiki					
Main Content	0.5	1	0.67	0.63	0.96	0.76
Bukan Main Content	1	0.91	0.95	1	0.95	0.97
Avg Total	0.96	0.92	0.93	0.97	0.95	0.95
Accuracy	0.916312			0.949060		

sebuah blok atau segmen sehingga model dapat mengenali dan memprediksi sebuah blok atau segmen dengan lebih baik.

2. Dengan melakukan dataset balancing maka permasalahan mengenai distribusi yang tidak seimbang antara label bukan *main content* dengan label *main content* yang dapat memberikan efek over-fitting pada model yang terbentuk dapat dihindari.
3. Dengan menganggap bahwa data yang memiliki permasalahan dari tahap pengambilan main content dengan template-based sebagai noisy data dan melakukan filterisasi terhadap data tersebut, maka data yang digunakan akan lebih representatif terhadap blok yang dihasilkan sehingga model yang terbentuk akan lebih akurat dalam melakukan prediksi.

Akan tetapi model yang terbentuk dari data yang telah dilakukan filterisasi terhadap permasalahan dari tahap pengambilan *main content* dengan menggunakan pendekatan *template-based* merupakan model yang terbentuk dengan bagian dari data dan bukan keseluruhan data. Sehingga diperlukan solusi untuk mengatasi permasalahan-permasalahan tersebut agar model yang terbentuk benar-benar representatif dengan keadaan yang sesungguhnya. Berberapa solusi yang dapat digunakan untuk mengatasi permasalahan-permasalahan tersebut diantaranya adalah:

1. Mengoptimalkan *threshold* atau batas minimal kesamaan *node* sehingga sebuah blok yang seharusnya dapat diproses lebih mendalam atau merupakan *template* sehingga dianggap unik karena blok atau segmen tersebut tidak muncul pada satu halaman web dapat dapat diproses lebih lanjut pada tahap pengambilan *main content* menggunakan pendekatan *template-based*.
2. Melakukan load terhadap *resource* yang dibutuhkan ketika ditemukan blok dengan *main content* yang menggunakan teknologi *embedded* atau iframe .
3. Menggunakan algoritma *text similarity* untuk membandingkan *atribute node* sehingga blok dengan *main content* yang memiliki *attribute Node* atau HTML *tag* yang berubah secara dinamis dapat diproses lebih lanjut pada tahap pengambilan *main content* menggunakan pendekatan *template-based*.

Diharapkan dengan memperbaiki permasalahan yang muncul dari temuan-temuan tersebut maka blok yang merupakan *main content* dapat lebih representatif terhadap keseluruhan data sehingga akurasi dari model klasifikasi yang telah terbentuk menjadi lebih akurat sesuai dengan kenyataan yang ada terutama untuk mengambil *main content*.

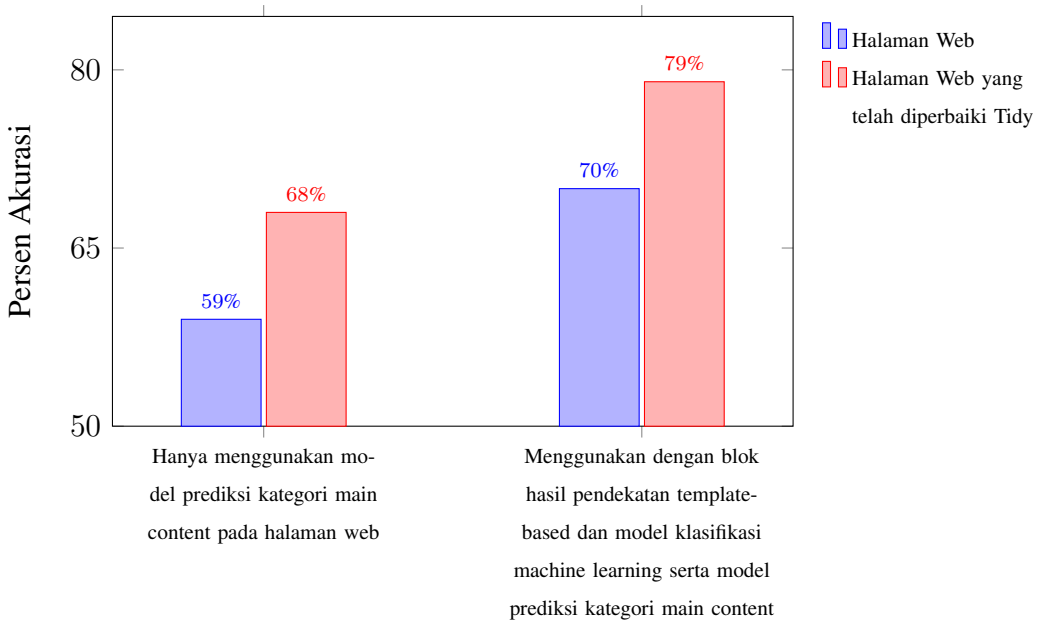
4.2.3 Pengujian dengan menggunakan blok hasil pendekatan *template-based*, model klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi dan model prediksi kategori *main content* pada halaman web pemerintah daerah yang dibangun oleh Wisnu [Sugiyanto, 2017]

Pengujian ketiga yaitu pengujian mengenai pengambilan *main content* dengan blok hasil pendekatan *template-based*, model klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi dan model prediksi kategori *main content* pada halaman web pemerintah daerah yang dibangun oleh Wisnu [Sugiyanto, 2017].

Pengujian ini dilakukan dengan menggunakan hasil yang didapatkan dari pengujian kedua dengan mengambil data yang diprediksi merupakan *main content* oleh model yang telah dibangun yang kemudian dilakukan klasifikasi terhadap kategori *main content* dengan menggunakan model yang telah di bangun pada peneli-

tian yang dilakukan oleh wisnu. Hasil yang didapatkan dari pengujian dapat dilihat pada gambar 4.34.

Dari gambar 4.34, terlihat bahwa dengan menggunakan kombinasi antara blok hasil pendekatan *template-based*, model klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi dan model prediksi kategori *main content* pada halaman web pemerintah daerah yang dibangun oleh Wisnu [Sugiyanto, 2017] akurasi prediksi untuk halaman web meningkat dari 59% ketika hanya menggunakan model prediksi kategori halaman web menjadi 68% ketika menggunakan kombinasi antara blok hasil pendekatan *template-based*, model klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi dan model prediksi kategori *main content* pada halaman web pemerintah daerah yang dibangun oleh Wisnu [Sugiyanto, 2017].



Gambar 4.34: Hasil Pengujian Prediksi Kategori Halaman Web

Pada halaman web yang telah diperbaiki tidy juga mengalami peningkatan akurasi dari 7-% ketika hanya menggunakan model prediksi kategori halaman web menjadi 79% ketika menggunakan kombinasi antara kombinasi antara blok hasil pendekatan *template-based*, model klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi dan model prediksi kategori *main content* pada halaman web pemerintah daerah yang dibangun oleh Wisnu [Sugiyanto, 2017].

Hal dapat menjadi faktor atas terjadinya peningkatan akurasi ini adalah model prediksi kategori halaman web yang dibangun oleh wisnu hanya menggunakan masukan berupa teks dari berberapa halaman web dan hanya menghapus berberapa karakter yang sering muncul pada halaman web sehingga bisa saja terdapat kata-kata yang bukan merupakan *main content* yang ikut terproses pada saat dilakukan prediksi kategori halaman web. Dengan permasalahan yang muncul tersebut dapat menurunkan tingkat akurasi dalam proses prediksi kategori halaman web.

Hal ini berbeda dengan hasil dari penelitian yang dilakukan ini, dimana dengan menggunakan pendekatan *template-based* dan model klasifikasi *machine learning* dengan *feature set* dari Yao yang telah dimodifikasi dapat mengurangi dan menghilangkan kata atau karakter yang bukan merupakan *main content* pada sebuah halaman web yang kemudian dilanjutkan dengan prediksi kategori halaman web. Hal yang perlu diingat bahwa akurasi masih dapat ditingkatkan lebih lanjut mengingat pada penelitian ini ditemukan berberapa permasalahan yang telah dibahas sebelumnya yang dapat mengurangi akurasi dalam pengambilan *main content*. Dengan mengatasi permasalahan-permasalahan tersebut maka kata-kata atau karakter yang didapat pada *main content* akan menjadi lebih representatif yang diharapkan dapat meningkatkan akurasi dari model yang dibangun pada penelitian Wisnu.

Halaman ini sengaja dikosongkan

BAB 5

KESIMPULAN DAN SARAN

Sebagai penutup dari tesis ini akan disajikan kesimpulan dari hasil penelitian dan pembahasan pada bab sebelumnya. Kemudian, akan dijelaskan mengenai saran dan penelitian mendatang yang didapat dari hasil kesimpulan

5.1 Kesimpulan

Berdasarkan pembahasan yang telah dilakukan pada bab 4, Berikut ini akan dibahas mengenai berberapa kesimpulan yang didapatkan.

5.1.1 Kesalahan Struktur Halaman Web Pemerintah Daerah

Dari hasil yang didapatkan pada tahap *preprocessing*, didapatkan bahwa sebagian besar halaman web yang ada pada situs web resmi pemerintah daerah memiliki permasalahan pada struktur HTML yang digunakan dengan *rule* untuk validasi struktur halaman web sesuai dengan tabel 4.2 dan menggunakan *validator* yang disediakan oleh w3c. Hal ini terlihat dari total 3267 halaman web yang dilakukan validasi, hanya 694 halaman web yang memenuhi persyaratan pada *rule* yang telah dibuat.

Hal ini menunjukkan bahwa web *developer* atau pihak yang bertanggung jawab dalam membangun situs resmi pemerintah daerah di Indonesia masih belum sepenuhnya mematuhi atau menggunakan panduan yang dikembangkan oleh W3C. Permasalahan terbesar yang paling sering muncul adalah mengenai *Mixed-up tags* seperti yang terlihat pada tabel 4.2, dimana menunjukkan penggunaan *tag* yang kurang benar. Hal seperti ini mungkin tidak nampak di *browser*, karena *browser* mungkin saja memperbaiki atau tidak menampilkan hal tersebut, akan tetapi hal ini akan sangat berpengaruh pada saat pembuatan node dari sebuah halaman web.

Pada penelitian ini juga dilakukan percobaan untuk memperbaiki halaman web yang tidak memenuhi hasil validasi dengan menggunakan aplikasi tidy, dimana menghasilkan 1574 halaman web yang memenuhi persayaratan pada *rule* yang telah dibuat. Hal ini menunjukkan bahwa walaupun telah dicoba dilakukan perbaikan dengan bantuan aplikasi atau *machine-assisted*, tidak semua halaman web dapat

memenuhi persyaratan dari *rule* yang telah dibuat.

Dengan demikian, masih diperlukan bantuan manusia untuk dapat memperbaiki permasalahan yang ada pada halaman web tersebut. Untuk kedepannya, hal ini dapat dicegah dengan melakukan pelatihan atau panduan mengenai standar atau panduan yang dikembangkan oleh W3C untuk web *developer* atau pihak yang bertanggung jawab dalam membangun situs resmi pemerintah daerah.

5.1.2 Pengambilan *main content* dengan menggunakan pendekatan Template-Based

Pada tahap pengambilan *main content* dengan menggunakan pendekatan *template-based*, hasil dari pengujian menunjukkan bahwa *main content* dapat terambil sepenuhnya dari setiap halaman web yang diujikan. Akan tetapi pendekatan *template-based* masih menghasilkan blok yang bukan *main content* dengan jumlah yang signifikan.

Tingginya jumlah blok yang bukan *main content* disebabkan oleh permasalahan mengenai perbedaan *layout* yang kecil pada berberapa halaman web yang ada pada satu situs pemerintah daerah dimana jika terdapat satu halaman web yang memiliki *layout* yang sedikit berbeda maka akan meningkatkan jumlah blok yang bukan *main content*. Hal ini dikarenakan pada penelitian ini digunakan nilai *threshold* atau batas minimal kesamaan *node* yang digunakan pada saat proses pendeteksian *template* adalah sebanyak jumlah halaman web yang diproses. Sebagai contoh jika sebuah situs resmi pemerintah daerah memiliki 9 halaman web yang telah lolos validasi yang kemudian akan diproses pada tahap pengambilan *main content* menggunakan pendekatan *template-based*.

Apabila terdapat sebuah blok yang muncul pada 8 halaman web dari 9 halaman web yang diproses tersebut maka blok tersebut dianggap sebagai sebuah blok yang unik pada 8 halaman web tersebut. Bergantung terhadap struktur HTML dari halaman web yang ada permasalahan tersebut dapat menimbulkan 2 hasil dimana 8 blok tersebut dianggap menjadi blok unik yang terpisah pada 9 halaman web tersebut dan menambah jumlah blok atau segmen yang bukan *main content* atau *parent* dari blok tersebut dianggap menjadi blok unik pada 9 halaman web tersebut

yang mengakibatkan *feature set* dari blok *main content* berubah menyerupai blok yang bukan *main content*. Hal ini dapat diatasi dengan mengoptimalkan *threshold* atau batas minimal kesamaan *node* sehingga sebuah blok yang seharusnya dapat diproses lebih mendalam atau merupakan *template* sehingga dianggap unik karena blok atau segmen tersebut tidak muncul pada satu halaman web dapat dapat diproses lebih lanjut pada tahap pengambilan *main content* menggunakan pendekatan *template-based*.

Selain itu juga terdapat permasalahan mengenai tag HTML yang dinamis atau berubah pada halaman web pemerintah daerah. Permasalahan ini umumnya muncul pada situs resmi pemerintah daerah yang menggunakan *Content Management System* dalam membangun situs mereka. Lebih lanjut, sekitar 90% dari situs yang memiliki permasalahan ini pada saat penelitian ini dilakukan adalah situs resmi yang dibangun dengan menggunakan *Content Management System* Wordpress sedangkan sisanya adalah situs resmi yang dibangun oleh Drupal dan Joomla. Sebagai gambaran, menurut penelitian yang dilakukan Aini [Nur Aini Rakhmawati, 2018], setidaknya terdapat 334 pemerintah daerah yang mengembangkan situs web resmi mereka menggunakan *Content Management System*.

Sebagai perbandingan, setidaknya 52.1% dari total situs web menggunakan *Content Management System* dalam membangun situs web mereka [w3tech, 2018b]. Selain itu Wordpress digunakan setidaknya 31 % dari total keseluruhan situs web di internet dan untuk pasar *Content Management System* sendiri, wordpress memiliki 59.8% *market share*. Hal ini menunjukkan bahwa, *Content Management System* adalah pilihan yang populer dalam membangun situs web untuk para pemilik situs web yang juga termasuk web developer atau pihak yang membangun situs web resmi pemerintah daerah. Berbagai alasan yang mungkin mempengaruhi keputusan *web developer* atau pihak yang membangun situs web resmi pemerintah daerah diantaranya adalah kemudahan dan tingkat keamanan yang sudah teruji. Pada penelitian yang dilakukan ini, situs web yang dibangun dengan *content management system* kurang dapat dilakukan pengambilan halaman web dengan sempurna sedangkan untuk situs web yang dibangun dari *scratch* atau memiliki HTML tag

yang *statis* akan dapat dilakukan pengambilan *main content* dengan sangat baik.

Sehingga untuk penelitian kedepannya perlu dilakukan pemahaman mengenai struktur atau cara kerja dari *Content Management System* dalam menyajikan *main content* pada halaman web agar akurasi pada pengambilan *main content* menggunakan *template-based* dapat ditingkatkan lebih jauh. Solusi lain yang dapat dilakukan adalah menggunakan algoritma *text similarity* untuk membandingkan *atribute node* sehingga blok dengan *main content* yang memiliki *Attribute Node* atau HTML tag yang berubah secara dinamis dapat diproses lebih lanjut pada tahap pengambilan *main content* menggunakan pendekatan *template-based*.

5.1.3 pengambilan *main content* melalui pendekatan klasifikasi *machine learning*

Pada tahap pengambilan *main content* melalui pendekatan klasifikasi *machine learning* dengan menggunakan *feature set* yang telah dimodifikasi dari yao untuk blok yang dihasilkan pada tahap pengambilan *main content* dengan menggunakan pendekatan *template-based*, menunjukkan bahwa model awal yang terbentuk masih belum dapat memprediksi blok yang merupakan *main content* dengan baik.

Hal tersebut dipengaruhi oleh permasalahan-permasalahan mengenai blok atau segmen yang diproses dengan kurang sempurna pada tahap *template-based* dan *dataset* yang terbentuk memiliki distribusi yang tidak seimbang. Permasalahan mengenai blok atau segmen yang diproses dengan kurang sempurna pada tahap template-based contohnya seperti perbedaan *layout* yang kecil pada berberapa halaman web atau tag HTML yang dinamis atau berubah pada halaman web pemerintah daerah, dan permasalahan mengenai *main content* yang terenkapsulasi sehingga kurang representatif. Berbagai permasalahan tersebut membuat *feature set* yang dimiliki oleh blok yang merupakan *main content* sangat mirip dengan blok yang bukan merupakan *main content* sehingga mengurangi akurasi pada model yang telah terbentuk. Solusi-solusi yang dapat dilakukan untuk mengatasi permasalahan-permasalahan tersebut adalah

1. Mengoptimalkan *threshold* atau batas minimal kesamaan *node* sehingga sebuah blok yang seharusnya dapat diproses lebih mendalam atau merupak-

an *template* sehingga dianggap unik karena blok atau segmen tersebut tidak muncul pada satu halaman web dapat dapat diproses lebih lanjut pada tahap pengambilan *main content* menggunakan pendekatan *template-based*.

2. Menggunakan algoritma *text similarity* untuk membandingkan *atribute node* sehingga blok dengan *main content* yang memiliki *Attribute Node* atau HTML tag yang berubah secara dinamis dapat diproses lebih lanjut pada tahap pengambilan *main content* menggunakan pendekatan *template-based*
3. Melakukan *load* terhadap *resource* yang dibutuhkan ketika ditemukan blok dengan *main content* yang menggunakan teknologi *embedded* atau iframe .

Berberapa hal juga dilakukan pada penelitian ini untuk mendapatkan model klasifikasi yang lebih baik dibandingkan dengan model awal yang dibentuk diantaranya adalah melakukan penambahan fitur, *dataset balancing* dan *filterisasi* terhadap dapat yang memiliki permasalahan pada tahap template-based. Penambahan fitur dilakukan untuk menangkap pola lebih banyak dari dataset yang terkumpul. *Dataset balancing* dilakukan untuk menyeimbangkan distribusi antara label *main content* dan label bukan *main content* dan menghindari adanya *over-fitting* pada dataset. Terakhir, filterisasi dilakukan untuk melihat apakah dengan menganggap data yang memiliki permasalahan pada tahap *template-based* sebagai noisy data dan menghilangkan *noisy data* tersebut dapat meningkatkan hasil yang dimiliki oleh model yang terbentuk.

Hasil penelitian mengenai penambahan fitur menunjukkan bahwa dengan melakukan penambahan fitur dari yang semula hanya menggunakan 4 macam fitur (jumlah kata, jumlah kalimat, jumlah link dan *text density*) menjadi 11 macam fitur (jumlah kata, jumlah kalimat, jumlah link, *text density*, *Maximum Consecutive Word*, *Maximum Consecutive Word*, *Mean Consecutive Word*, *Max Occurence Word*, *Text formatting*, *Table formatting*, *List formatting*, *Paragraph formatting*) dapat meningkatkan hasil atau performa dari model yang terbentuk secara signifikan terutama untuk nilai *precision* dan *recall* dari label *main content* yang sangat penting mengingat tujuan dari model klasifikasi dibentuk adalah untuk mengindentifikasi blok atau segmen yang merupakan *main content* dari kandidat *main content*. Se-

lain itu penggunaan *dataset balancing* SMOTE-EEN dan filterisasi mengenai data yang bermasalah, juga dapat meningkatkan hasil atau performa yang diperoleh dari model yang terbentuk terutama mengenai *precision* dan *recall* pada label *main content*.

5.2 Saran dan Penelitian Selanjutnya

Pada penelitian telah diidentifikasi dan dibahas mengenai permasalahan-permasalahan yang muncul ketika penelitian dilakukan. Berbagai contoh saran dan solusi telah dijelaskan dan dibahas pada bagian hasil dan pembahasan untuk diteliti lebih lanjut pada penelitian selanjutnya. Saran dan solusi tersebut adalah:

1. Memberikan pelatihan atau panduan kepada pihak web *developer* atau atau pihak yang bertanggung jawab dalam membangun situs web resmi pemerintah daerah mengenai pembangunan situs berdasarkan standar yang telah dikembangkan oleh W3C.
2. Pihak pengurus situs perlu memperhatikan halaman web yang hanya terdiri atas sebuah link seperti link untuk men-*download* perda dimana sebaiknya dijadikan seluruh halaman web yang berbentuk seperti demikian dipusatkan menjadi sebuah halaman yang berisi daftar link yang ada.
3. Mengoptimalisasi nilai *threshold* atau batas minimal kesamaan *node* yang digunakan agar pendeteksian blok yang merupakan *template* dapat ditingkatkan.
4. Menggunakan algoritma *text similarity* untuk membandingkan *attribute node* agar dapat membandingkan *attribute node* yang mirip dan mengambil keputusan apakah *node* tersebut sama ataukah berbeda.
5. Melakukan *load* terhadap *resource* yang dibutuhkan ketika ditemukan blok atau segmen unik yang memiliki konten *embedded* atau iframe di dalamnya.

Saran dan solusi di atas diharapkan dapat meningkatkan akurasi dalam pengambilan *main content* pada halaman web untuk penelitian-penelitian selanjutnya yang menggunakan dasar dari penelitian yang telah dilakukan ini.

DAFTAR PUSTAKA

[AL and HB, 1992] AL, C. and HB, L. (1992). *A first course in factor analysis (2nd edition)*. Erlbaum, NJ:Hillsdale.

[Alarte et al., 2015] Alarte, J., Insa, D., Silva, J., and Tamarit, S. (2015). TeMex. In *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion*, pages 155–158, New York, New York, USA. ACM Press.

[Baluja and Shumeet, 2006] Baluja, S. and Shumeet (2006). Browsing on small screens. In *Proceedings of the 15th international conference on World Wide Web - WWW '06*, page 33, New York, New York, USA. ACM Press.

[Bar-Yossef and Rajagopalan, 2002] Bar-Yossef, Z. and Rajagopalan, S. (2002). Template detection via data mining and its applications. *Proceedings of the 11th international conference on World Wide Web*, pages 580–591.

[Barua et al., 2014] Barua, J., Patel, D., and Agrawal, A. K. (2014). Removing noise content from online news articles.

[Batista et al., 2004] Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29.

[Borges and Levene, 2000] Borges, J. and Levene, M. (2000). Data Mining of User Navigation Patterns. In *Data Mining of User Navigation Patterns*, pages 92–112. Springer, Berlin, Heidelberg.

[Cai et al., 2003] Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. (2003). VIPS: a Vision-based Page Segmentation Algorithm.

[Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

[Chen et al., 2003] Chen, Y., Ma, W.-Y., and Zhang, H.-J. (2003). Detecting web page structure for adaptive viewing on small form factor devices. *Proceedings of the 12th international conference on World Wide Web*, pages 225–233.

[dan Informasi, 2009] dan Informasi, D. K. (2009). Panduan Penyelenggaraan Situs Pemerintah Daerah.

[dan Informatika, 2003] dan Informatika, K. K. (2003). Instruksi Presiden Republik Indonesia Nomer 3 Tahun 2003 Mengenai Kebijakan Dan Strategi Nasional Pengembangan E-Government.

[Debnath et al., 2005] Debnath, S., Mitra, P., Pal, N., and Giles, C. L. (2005). Automatic Identification of Informative Sections of Web Pages. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1233–1246.

[Dewi and Mudjahidin, 2014] Dewi, L. A. S. and Mudjahidin (2014). Analisis Penerapan Aplikasi Surabaya Single Windows Pemerintah Kota Surabaya Menggunakan Government Adoption Model (GAM). *JURNAL TEKNIK POMITS*, 3(2):A210–A21.

[Etzioni and Oren, 1996] Etzioni, O. and Oren (1996). The World-Wide Web: quagmire or gold mine? *Communications of the ACM*, 39(11):65–68.

[Feeney and Brown, 2017] Feeney, M. K. and Brown, A. (2017). Are small cities online? Content, ranking, and variation of U.S. municipal websites. *Government Information Quarterly*, 34(1):62–74.

[Fornell and Larcker, 1981] Fornell, C. and Larcker, D. F. (1981). Structural equation models with unobservable variables and measurement error: Algebra and statistics. *Journal of marketing research*, pages 382–388.

[Friedman and Bryen, 2007] Friedman, M. G. and Bryen, D. N. (2007). Web accessibility design recommendations for people with cognitive disabilities. *Technology and Disability*, 19(4):205–212.

[Gao and Fan, 2014] Gao, B. and Fan, Q. (2014). Multiple Template Detection Based on Segments. pages 24–38. Springer, Cham.

[Gibson et al., 2005] Gibson, D., Punera, K., and Tomkins, A. (2005). The volume and evolution of web page templates. *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 830–839.

[Gibson et al., 2007] Gibson, J., Wellner, B., and Lubar, S. (2007). Adaptive web-page content identification. In *Proceedings of the 9th annual ACM international workshop on Web information and data management - WIDM '07*, page 105, New York, New York, USA. ACM Press.

[Gottron., 2009] Gottron., T. (2009). *Content Extraction: Identifying the Main Content in HTML Documents*. PhD thesis, Johannes Gutenberg-Universität.

[Hair JF and W, 1998] Hair JF, Tatham RL, A. R. and W, B. (1998). *Multivariate data analysis (Fifth Ed.)*. Prentice-Hall, London.

[Hermawan, 2015] Hermawan, D. P. (2015). Evaluasi Website Pemerintah Daerah Provinsi dan Kabupaten/Kota di Indonesia dengan Menggunakan Development Stage Model dan Peraturan Depkominfo. *Jurusan Sistem Informasi*.

[Hoens and Chawla, 2013] Hoens, T. R. and Chawla, N. V. (2013). Imbalanced datasets: from sampling to classifiers. *Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley*.

[Hoesin et al., 2008] Hoesin, H., Setiadi, H., Lemmung, N. A., Tonandriv, P. A., and Abdulloh (2008). Penilaian Situs Pemerintahan Daerah di Provinsi DKI Jakarta, Bengkulu, Jambi, dan Bangka Belitung.

[Huang and Benyoucef, 2014] Huang, Z. and Benyoucef, M. (2014). Usability and credibility of e-government websites. *Government Information Quarterly*, 31(4):584–595.

[Jeni et al., 2013] Jeni, L. A., Cohn, J. F., and De La Torre, F. (2013). Facing imbalanced data–recommendations for the use of performance metrics. In *Affective*

Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on, pages 245–251. IEEE.

[JP, 1992] JP, S. (1992). *Applied multivariate statistics for the social sciences (2nd edition)*. Erlbaum, NJ:Hillsdale.

[Kohlschütter et al., 2010] Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining - WSDM '10*, page 441, New York, New York, USA. ACM Press.

[Kosala and Blockeel, 2000] Kosala, R. and Blockeel, H. (2000). Web Mining Research: A Survey.

[Krishna and Dattatraya, 2015] Krishna, S. S. and Dattatraya, J. S. (2015). Schema inference and data extraction from templatized Web pages. In *2015 International Conference on Pervasive Computing (ICPC)*, pages 1–6. IEEE.

[Kulkarni et al., 2015] Kulkarni, H. H., Manasi, M., Kulkarni, K., and Professor, A. (2015). Template Extraction from Heterogeneous Web Pages. *International Journal of Electrical, Electronics and Computer Engineering*, 4(1):125–131.

[Kumar, 2015] Kumar, S. N. (2015). World towards Advance Web Mining: A Review. 3(2):44–61.

[Lin and Ho, 2002] Lin, S.-H. and Ho, J.-M. (2002). Discovering informative content blocks from Web documents. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, page 588, New York, New York, USA. ACM Press.

[Liu, 2011] Liu, B. (2011). *Web data mining : exploring hyperlinks, contents, and usage data*. Springer.

[Louvan, 2009] Louvan, S. (2009). EXTRACTING THE MAIN CONTENT FROM WEB DOCUMENTS.

[Luengo et al., 2011] Luengo, J., Fernández, A., García, S., and Herrera, F. (2011). Addressing data complexity for imbalanced data sets: analysis of smote-based oversampling and evolutionary undersampling. *Soft Computing*, 15(10):1909–1936.

[Lundgren et al., 2015] Lundgren, E., Papapetrou, P., and Asker, L. (2015). Extracting news text from web pages. In *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '15*, pages 1–4, New York, New York, USA. ACM Press.

[Mulus, 2009] Mulus, R. T. (2009). Analisis E-Government pada Kabupaten/Kota di Indonesia.

[Musso et al., 2000] Musso, J., Weare, C., and Hale, M. (2000). Designing web technologies for local governance reform: Good management or good democracy? *Political Communication*, 17(1):1–19.

[Nur Aini Rakhmawati, 2018] Nur Aini Rakhmawati, Sayekti Harits, D. H. M. a. F. (2018). A survey of web technologies used in indonesia local government. *JURNAL SISFO*, 7(3):213–222.

[Park et al., 2013] Park, T. H., Saxena, A., Jagannath, S., Wiedenbeck, S., and Forte, A. (2013). Towards a taxonomy of errors in html and css. In *Proceedings of the ninth annual international ACM conference on International computing education research*, pages 75–82. ACM.

[Peters and Lecocq, 2013] Peters, M. E. and Lecocq, D. (2013). Content extraction using diverse feature sets. In *Proceedings of the 22nd International Conference on World Wide Web - WWW '13 Companion*, pages 89–90, New York, New York, USA. ACM Press.

[Prakasam and Suresh, 2010] Prakasam, S. and Suresh, R. M. (2010). An agent -based Intelligent System to enhance E-Learning through Mining Techniques. *International Journal on Computer Science and Engineering*, 02(03):759–763.

[Raggett, 1998] Raggett, D. (1998). Clean up your web pages with hp's html tidy. *Computer networks and ISDN systems*, 30(1-7):730–732.

[Saito and Rehmsmeier, 2015] Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432.

[Samimi et al., 2012] Samimi, H., Schäfer, M., Artzi, S., Millstein, T., Tip, F., and Hendren, L. (2012). Automated repair of html generation errors in php applications using string constraint solving. In *Software Engineering (ICSE), 2012 34th International Conference on*, pages 277–287. IEEE.

[Sosiawan and Arief., 2008] Sosiawan, E. and Arief. (2008). Tantangan Dan Hambatan Dalam Implementasi eGovernment Di Indonesia.

[Sugiyanto, 2017] Sugiyanto, W. T. (2017). Klasifikasi Kelengkapan Konten Website dan Media Sosial Pemerintah Daerah di Indonesia dengan Menggunakan Metod Naïve-Bayes.

[Sulistyo et al., 2008] Sulistyo, D., Negara, H. P., and Firdaus, Y. (2008). Analisis Kajian Standarisasi Isi Situs Web.

[TasnimSiddiqui and Aljahdali, 2013] TasnimSiddiqui, A. and Aljahdali, S. (2013). Web Mining Techniques in E-Commerce Applications. *International Journal of Computer Applications*, 69(8):39–43.

[Vieira et al., 2006] Vieira, K., da Silva, A. S., Pinto, N., de Moura, E. S., Cavalcanti, J. M. B., and Freire, J. (2006). A fast and robust method for web page template detection and removal. *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 258–267.

[w3 Organization, 2018] w3 Organization (2018). html main element.

[W3C, 2017] W3C (2017). What is the Document Object Model?

[W3C, 2018] W3C (2018). Html tag.

[w3tech, 2018a] w3tech (2018a). Html tags ordered by category.

[w3tech, 2018b] w3tech (2018b). Usage of content management systems for websites.

[Weninger et al., 2010] Weninger, T., Hsu, W. H., and Han, J. (2010). CETR. In *Proceedings of the 19th international conference on World wide web - WWW '10*, page 971, New York, New York, USA. ACM Press.

[Yao and Zuo, 2013] Yao, J. and Zuo, X. (2013). A Machine Learning Approach to Webpage Content Exraction.

[Yi et al., 2003] Yi, L., Liu, B., and Li, X. (2003). Eliminating noisy information in Web pages for data mining. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, page 296, New York, New York, USA. ACM Press.

[Youngblood and Mackiewicz, 2012] Youngblood, N. E. and Mackiewicz, J. (2012). A usability analysis of municipal government website home pages in Alabama. *Government Information Quarterly*, 29(4):582–588.

[Yunis, 2016] Yunis, H. (2016). *Content Extraction from Webpages Using Machine Learning*. PhD thesis.

[Zeleny et al., 2017] Zeleny, J., Burget, R., and Zendulka, J. (2017). Box clustering segmentation: A new method for vision-based web page preprocessing. *Information Processing & Management*, 53(3):735–750.

Halaman ini sengaja dikosongkan